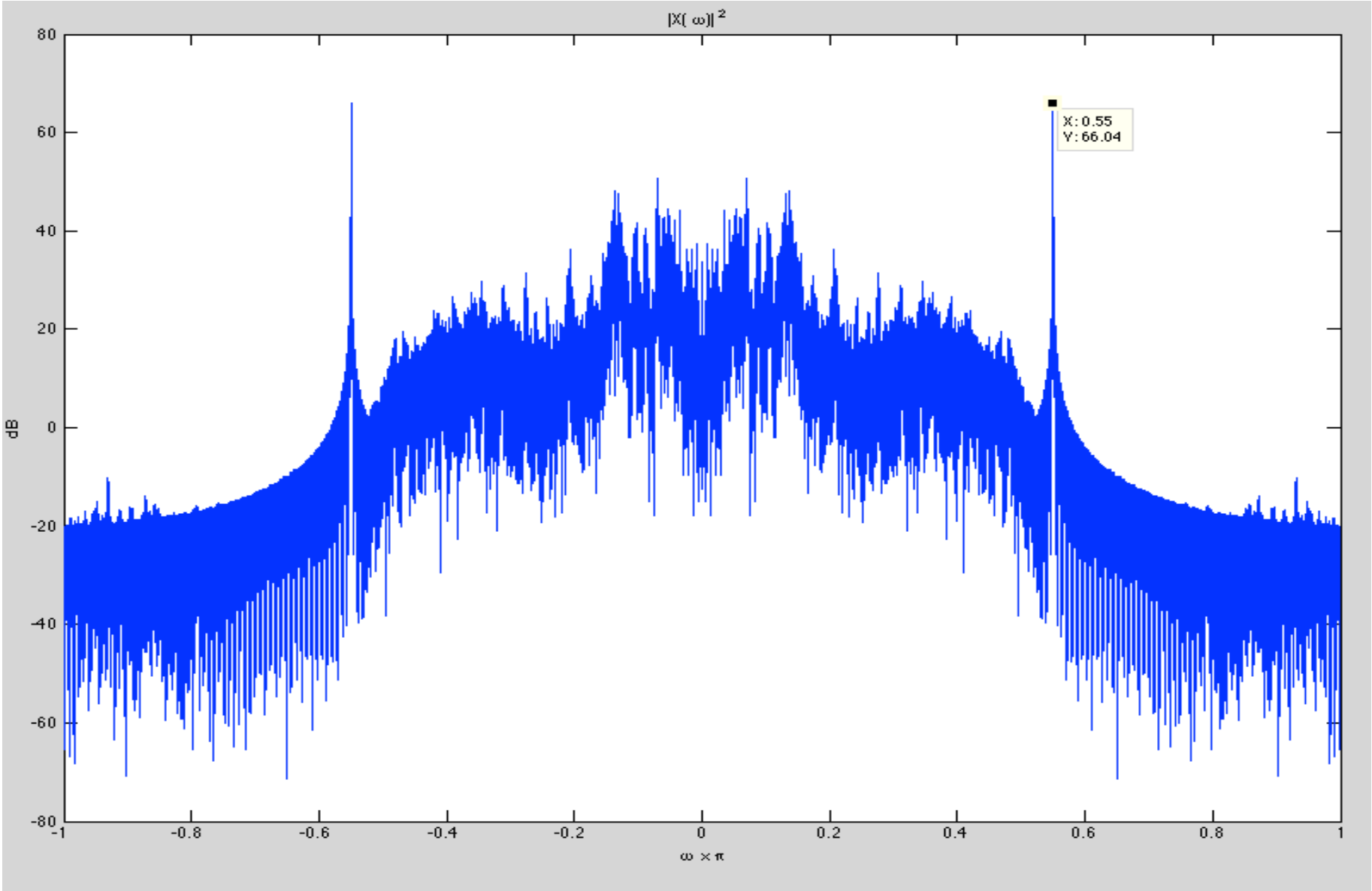
GEL-3003

TP #1 A2014

TP #1, Q1

```
% Nfft: nombre de points de la DFT (fonction fft)
Nfft = 200000;
deltaf = 2/Nfft
xx = [-1:deltaf:1-deltaf];
yy = 20*log10(abs(fftshift(fft(data,Nfft))));
figure, plot(xx,yy);
figure, plot(xx,yy);
xlabel(' \omega \times \pi'); ylabel('dB');
title('|X(\omega)|^2');


freqz(RIF_1,1);
freqz(RII_b,RII_a);
figure, zplane(RIF_1,1);
figure, zplane(RII_b,RII_a);
```

TFD du signal audio (module^2)



Résolution en fréquence vs précision

```
data = data1 + (0.1*cos(0.55*pi*(1:40100)))';
```

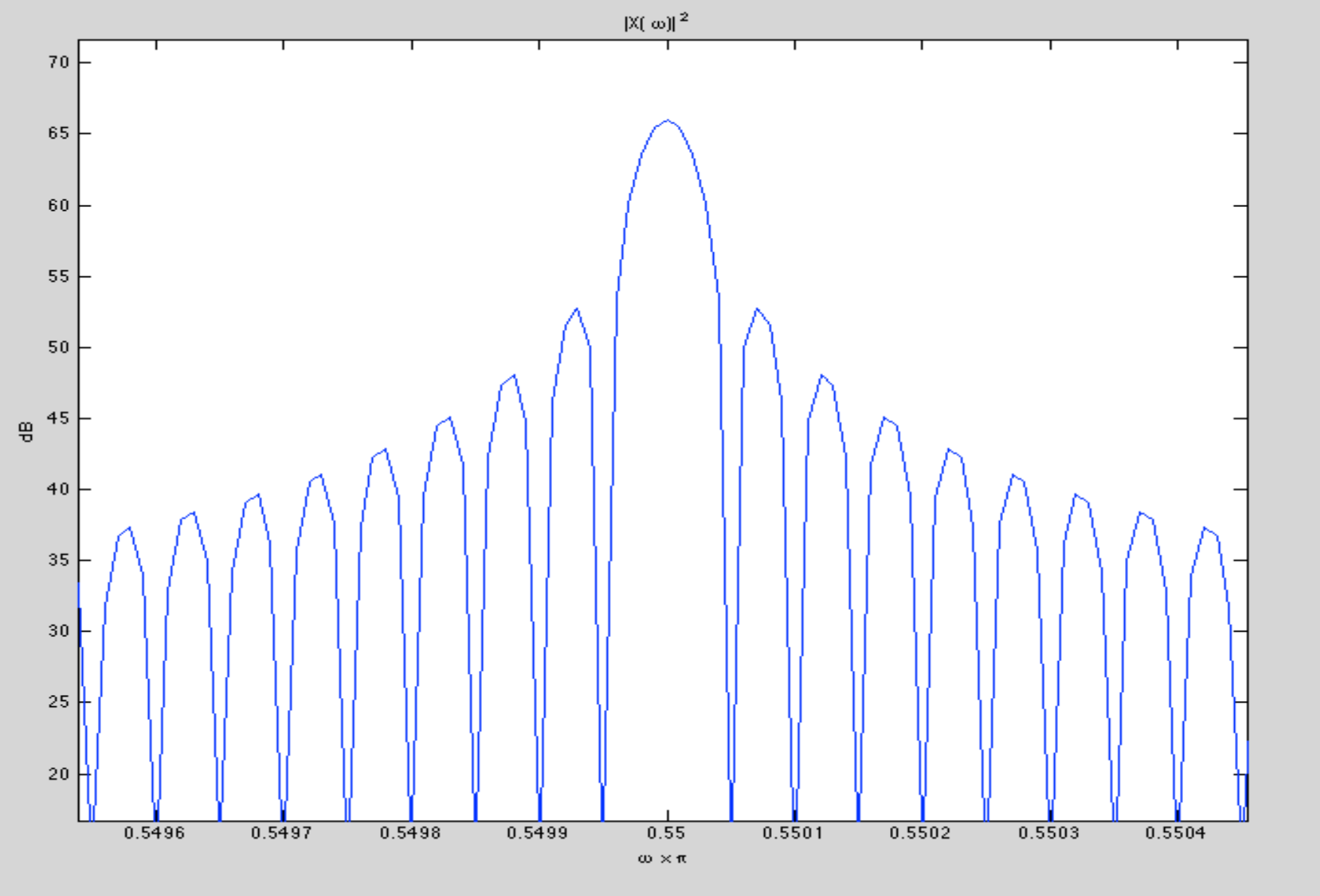
% Le signal a 40100 points; la résolution en fréquence est déterminée par le nombre de points du signal et la fenêtre qui est utilisée

Résolution = demi-largeur du lobe principal
= $2\pi/40100 = 5\times10^{-5}\ \pi$ pour la fenêtre rectangulaire

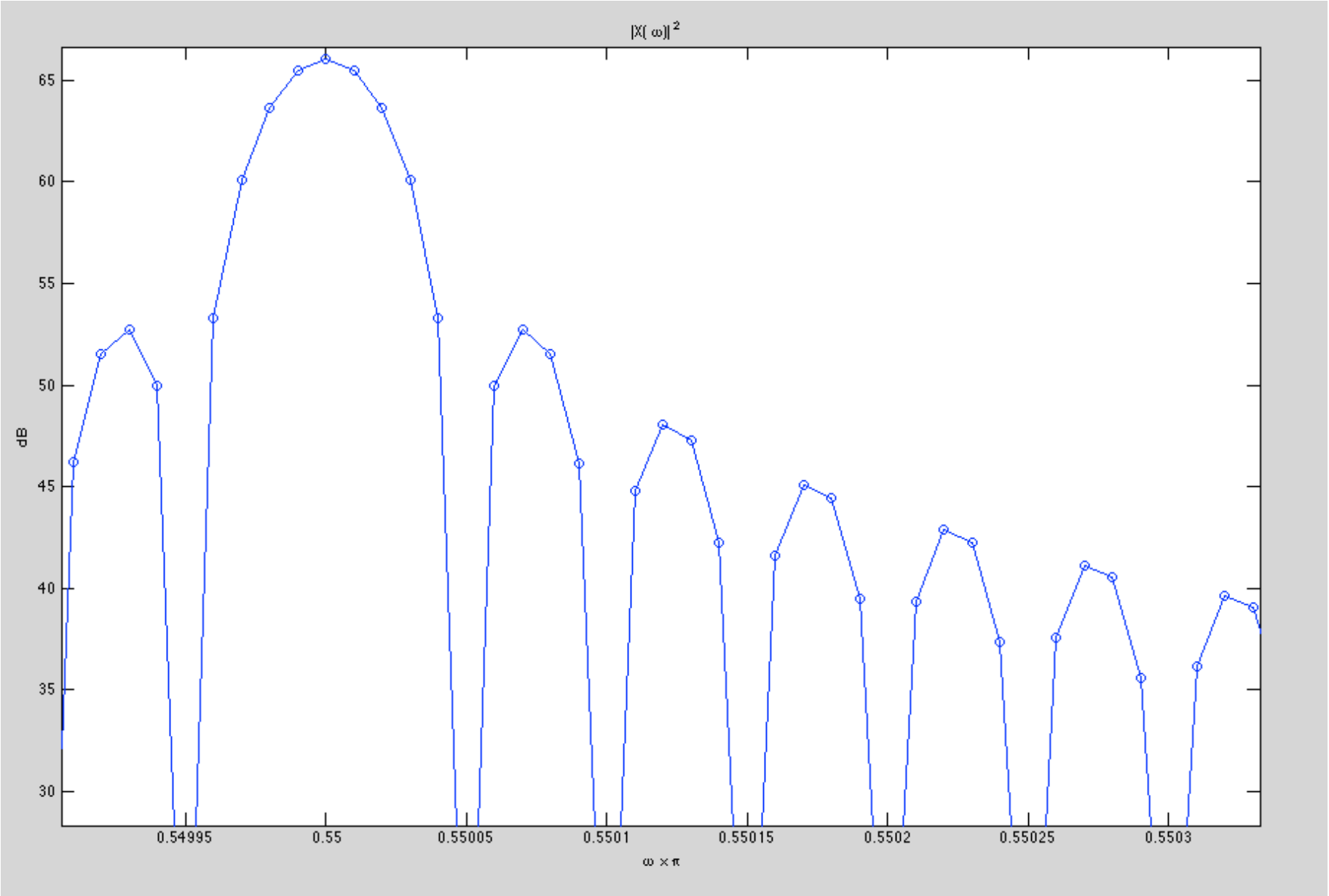
% A) TFD (fft) calculée sur 200000 points; il y a donc un point de la DFT à chaque $2\pi/200000 = 10^{-5}\ \pi$

% B) TFD (fft) calculée sur 50000 points; il y a donc un point de la DFT à chaque $2\pi/50000 = 4\times10^{-5}\ \pi$

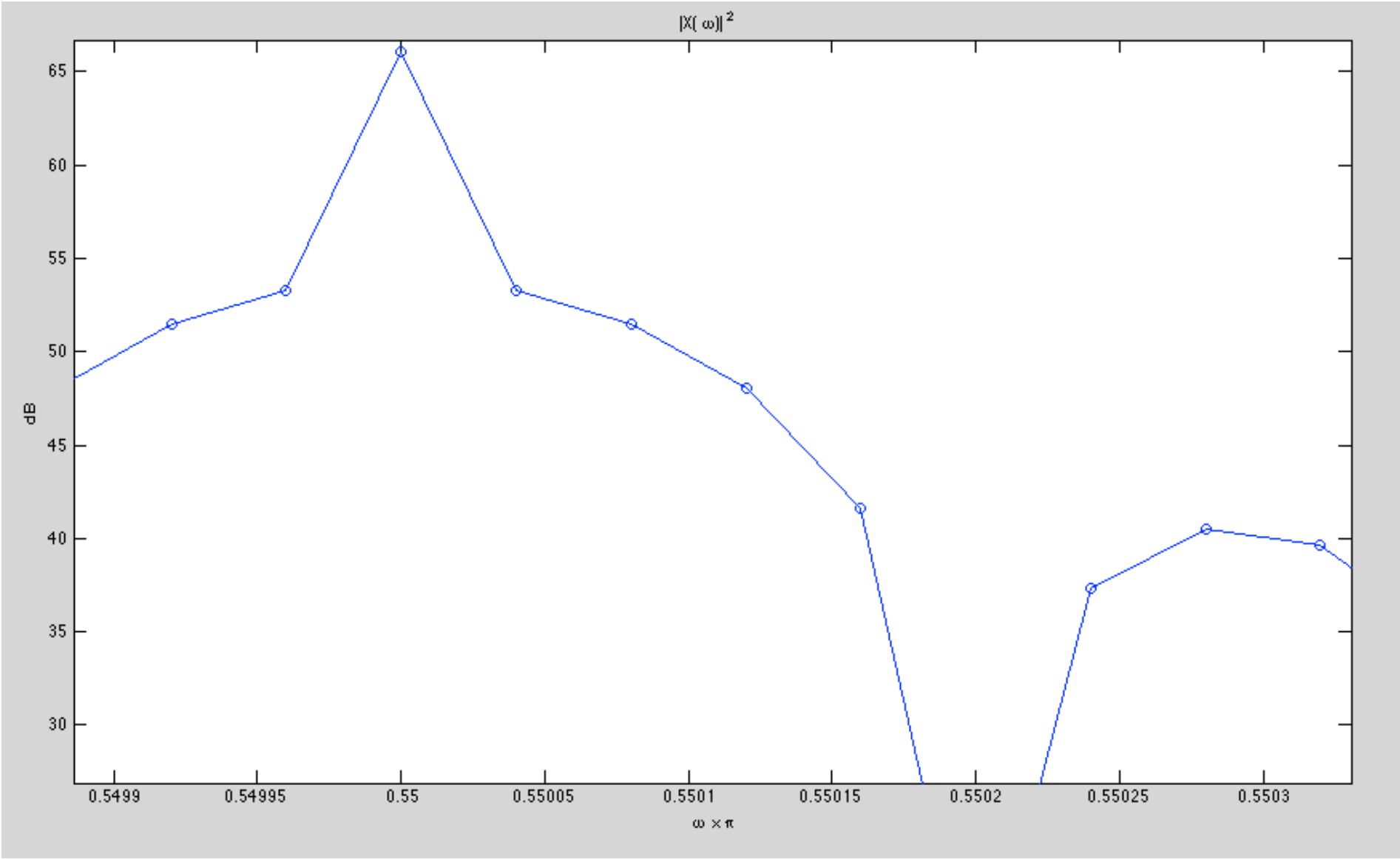
L = 40100, fen. Rectang, Nfft = 200000



L = 40100, fen. Rectang, Nfft = 200000



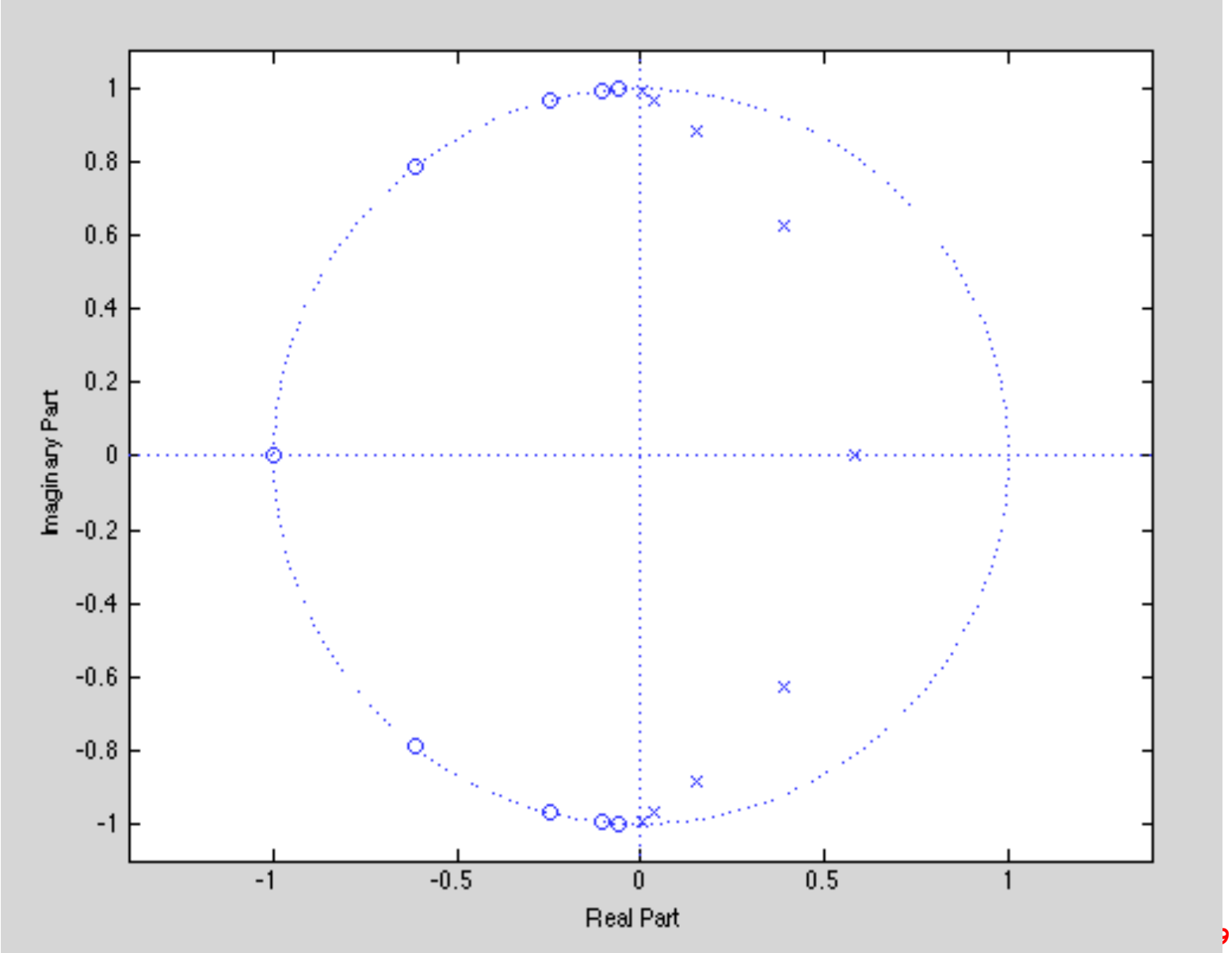
L = 40100, fen. Rectang, Nfft = 50000



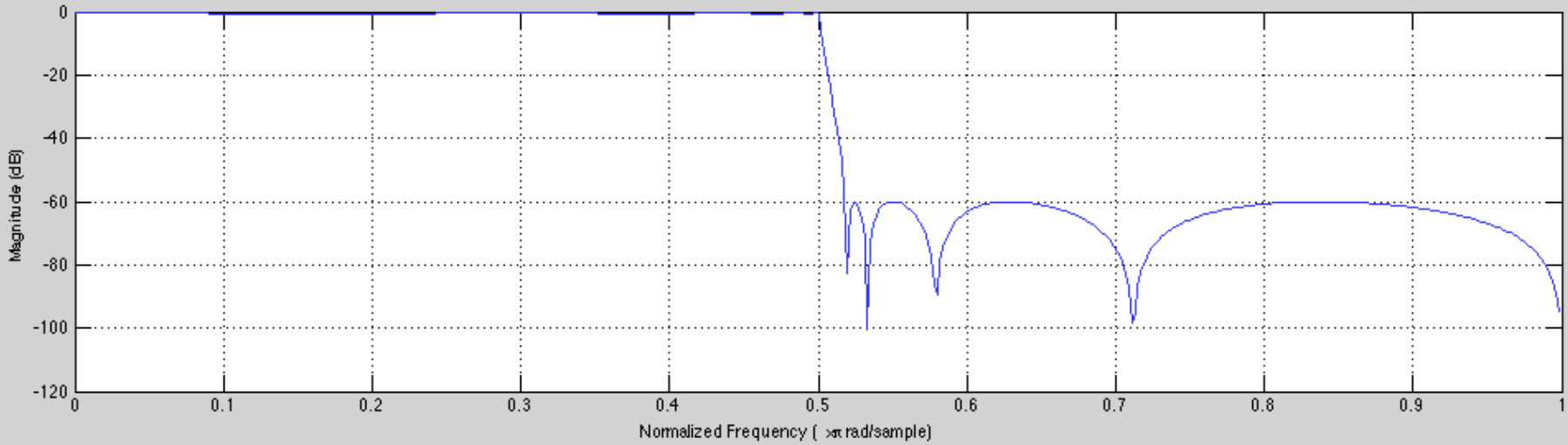
TP #1, Q1

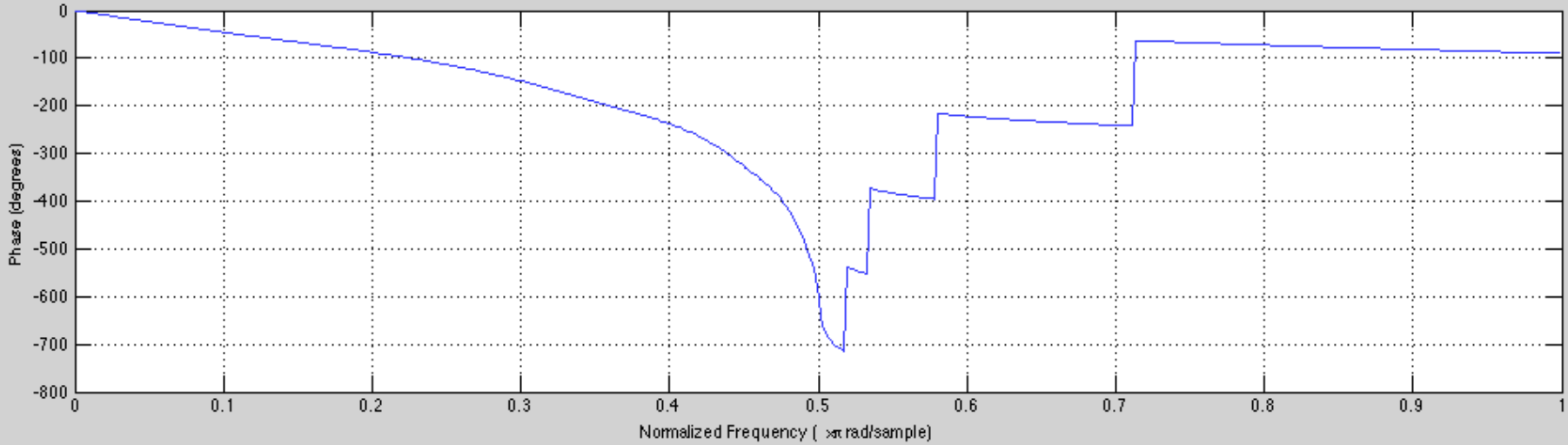
```
figure, zplane(RIF_1,1);
freqz(RIF_1,1);

figure, zplane(RII_b,RII_a);
freqz(RII_b,RII_a);
```

Filtre RII

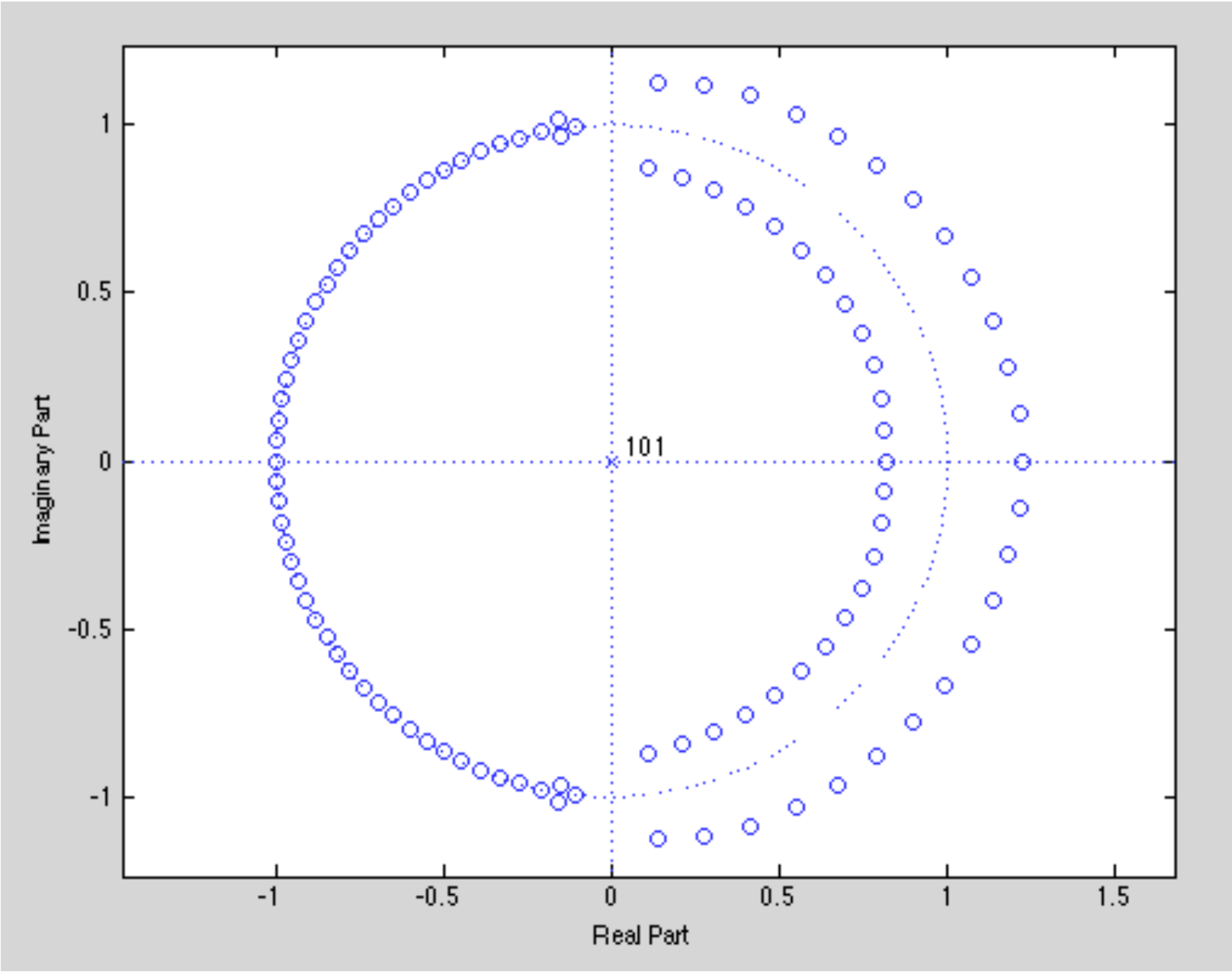


Filtre RII ordre 10

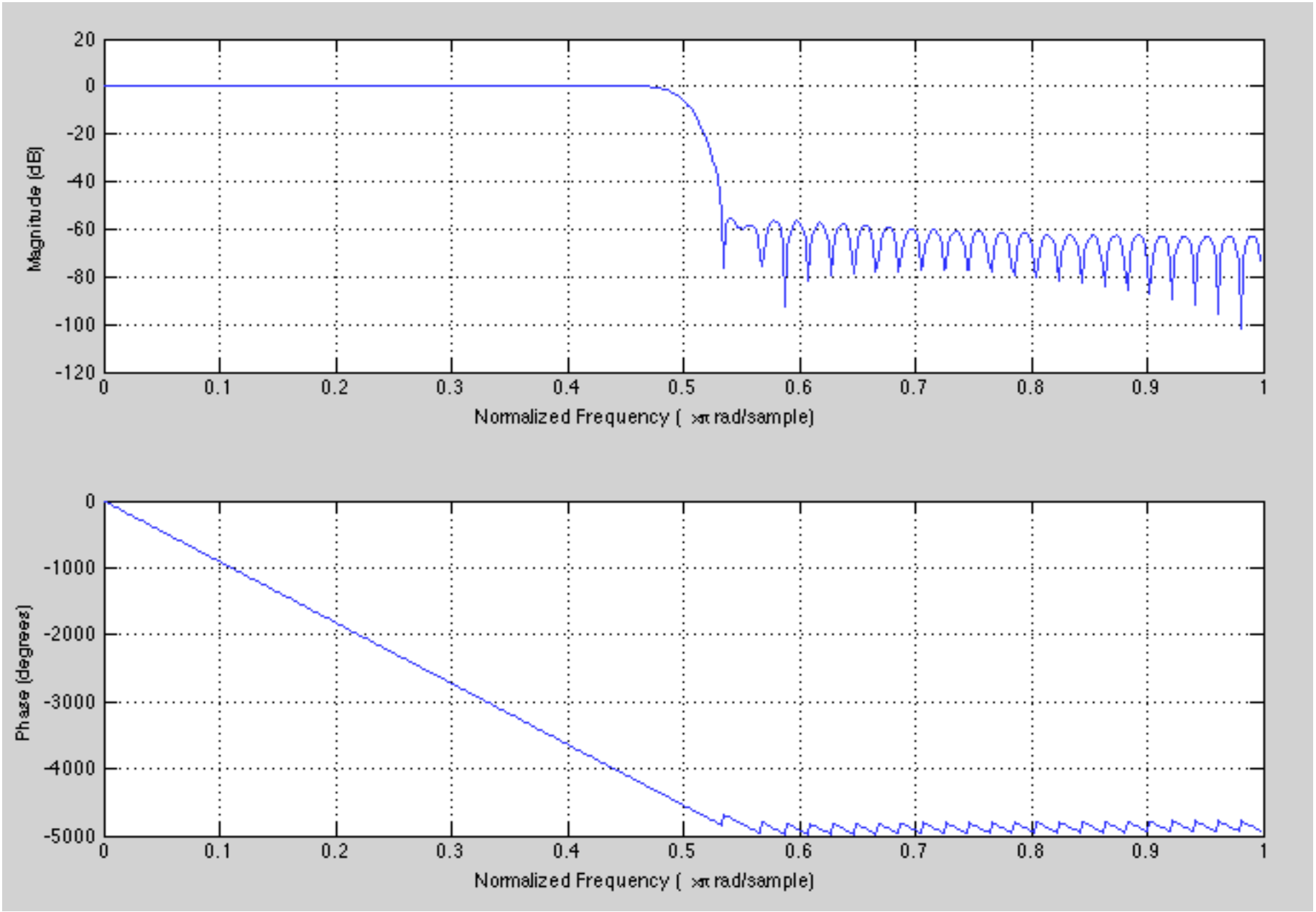




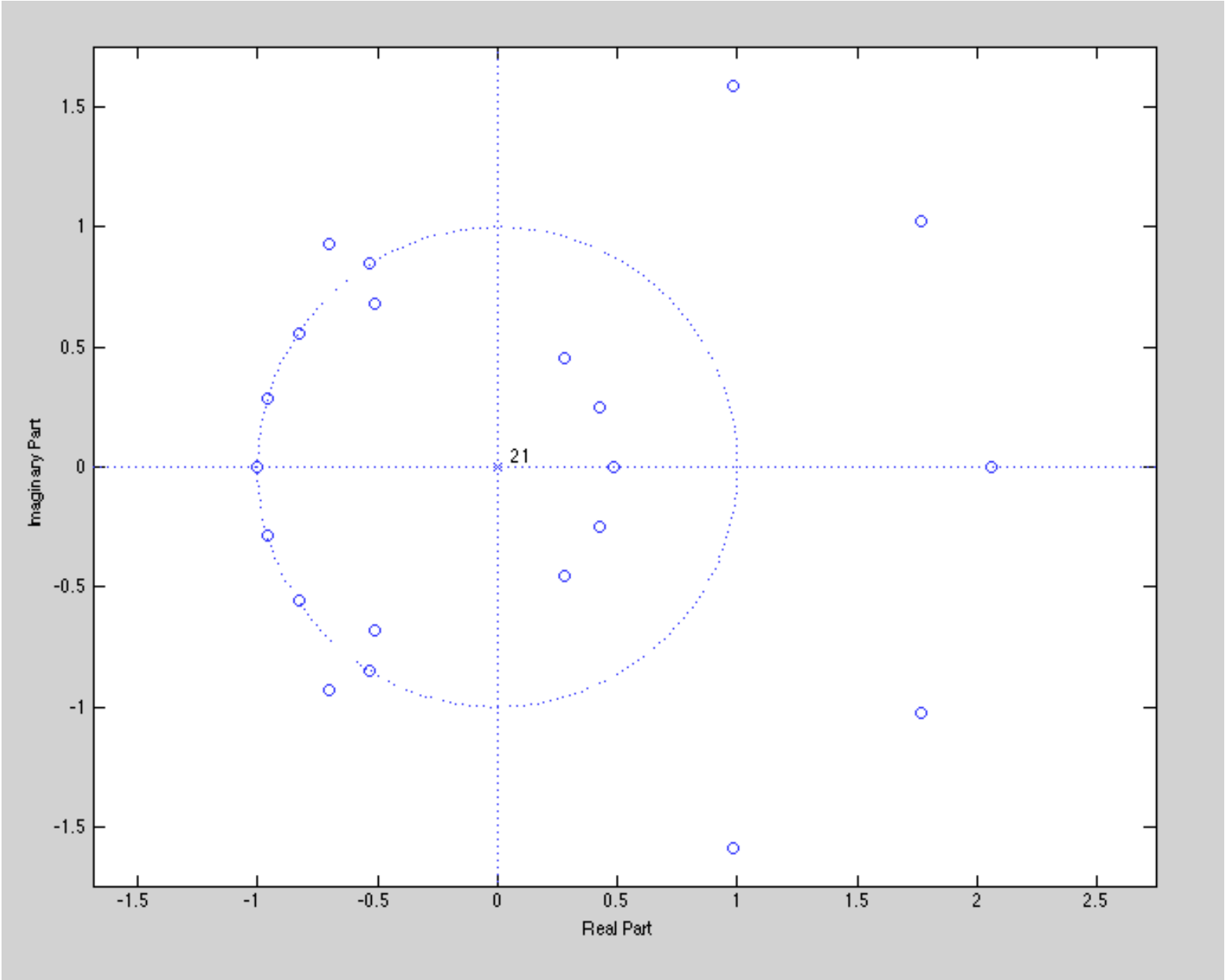
Filtre RIF_1



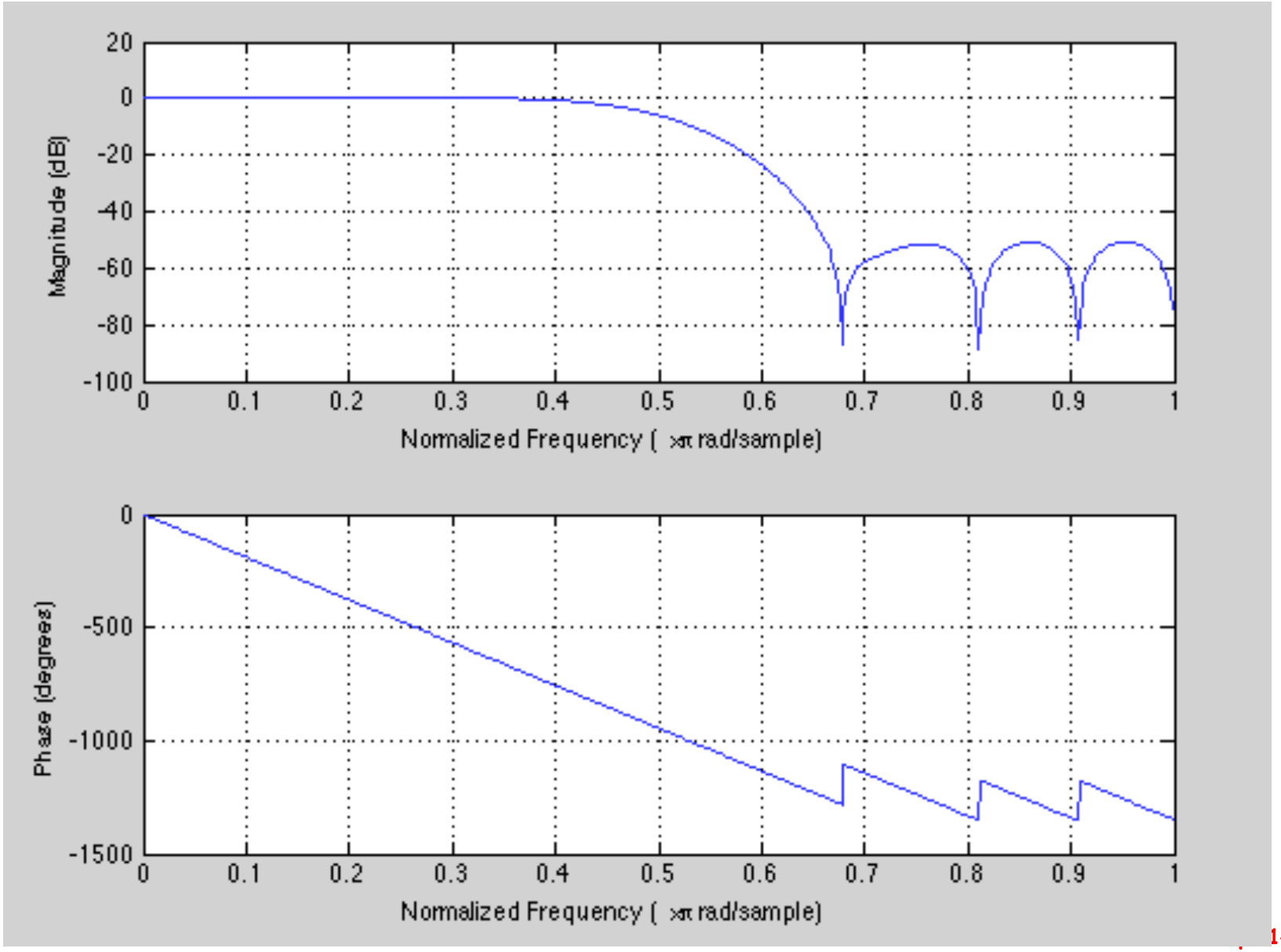
Filtre RIF avec 102 coefficients



Filtre RIF_2



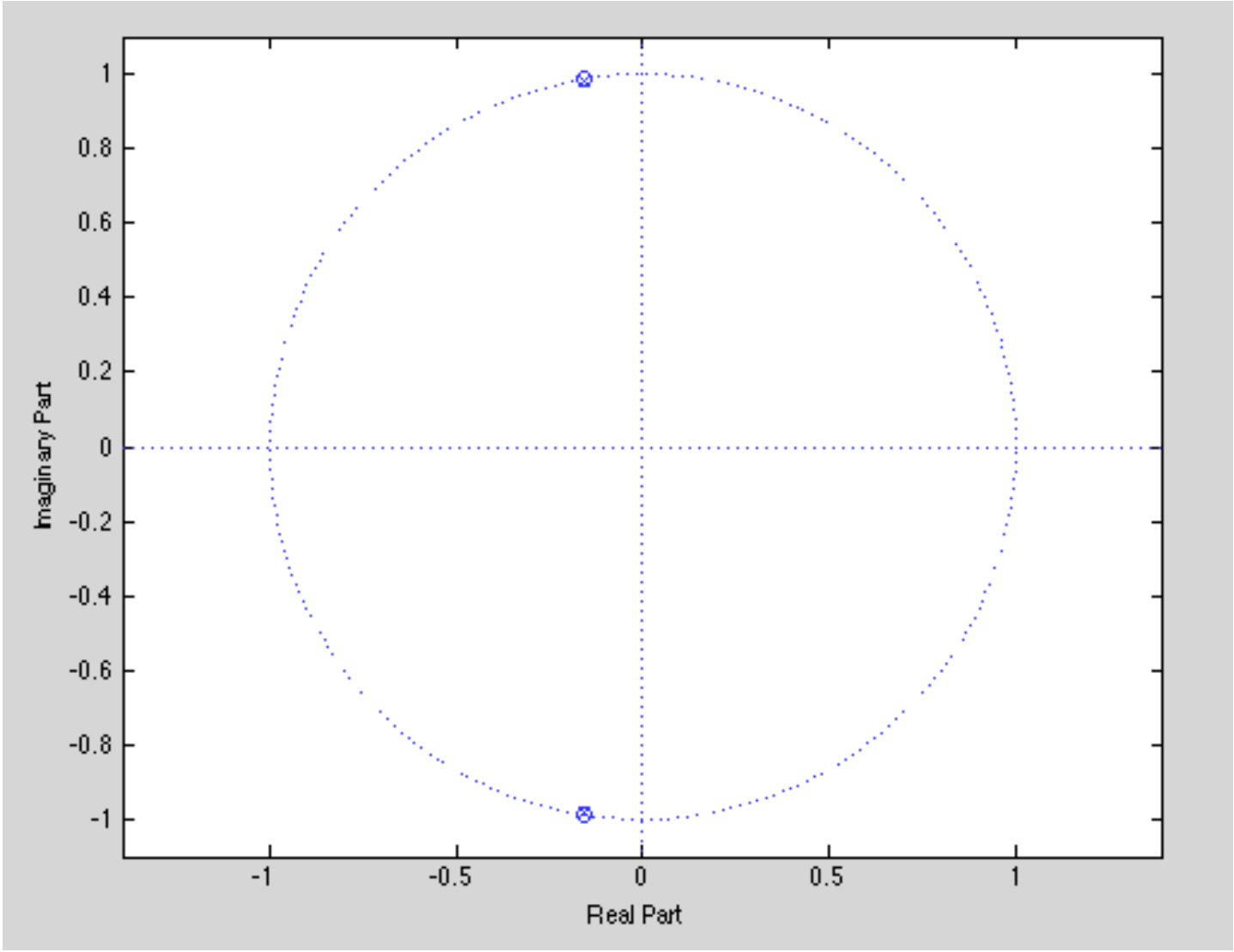
Filtre RIF avec 22 coefficients



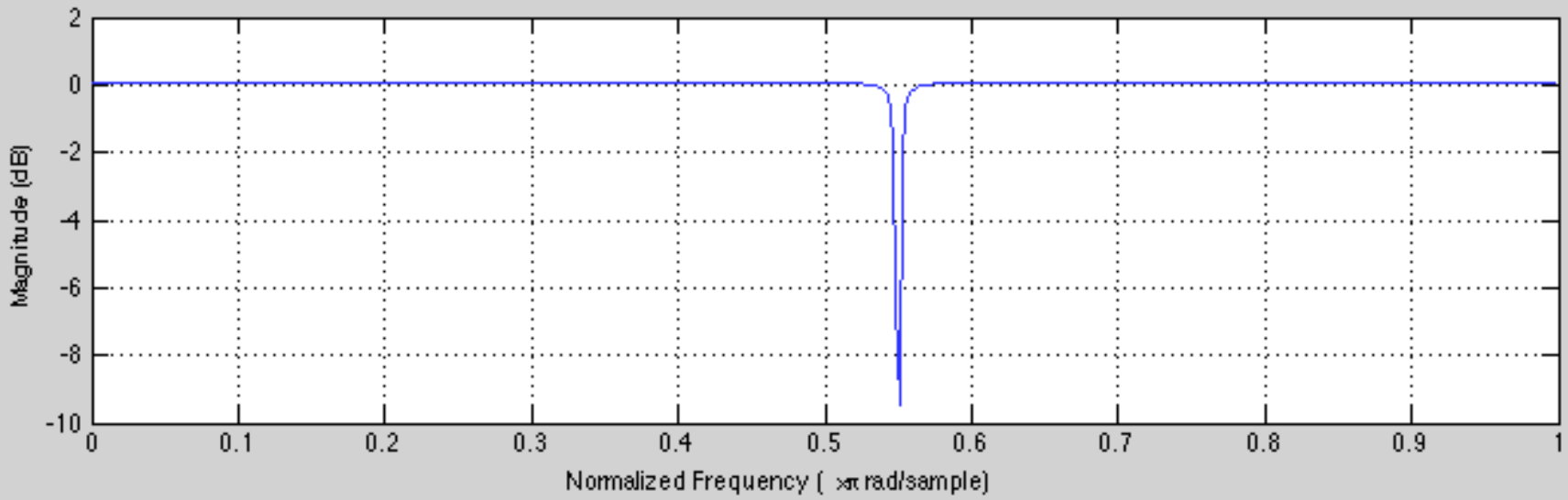
TP #1, Q1 (suite)

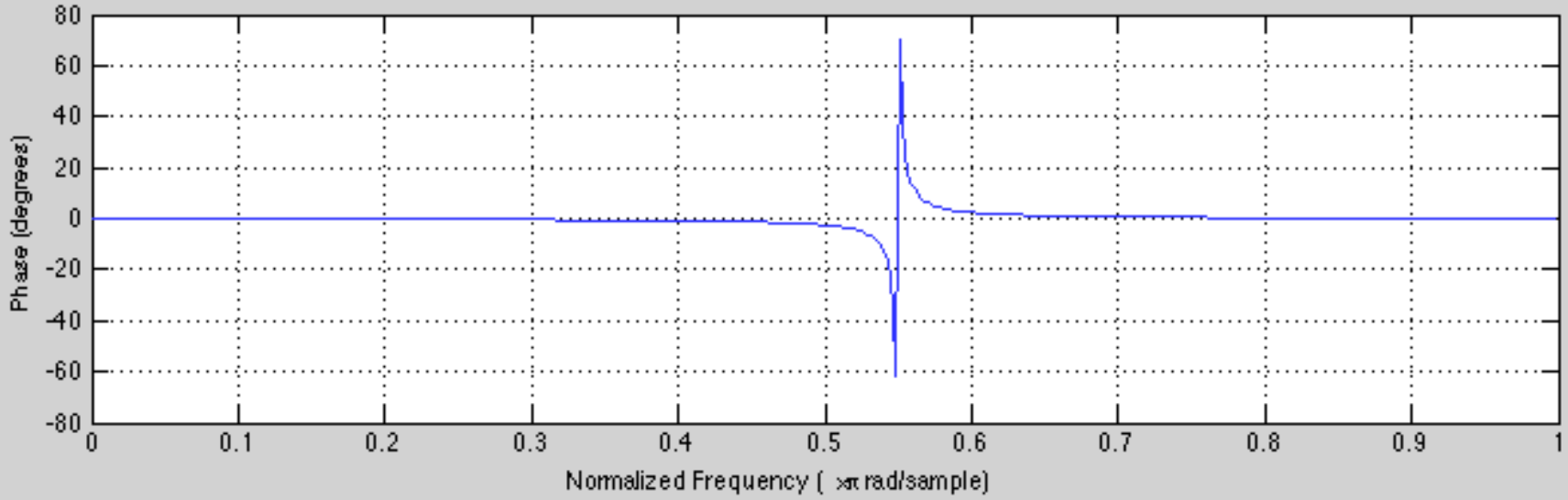
```
% filtrage du signal musical
% filtre RIF
data_fir1 = conv(data,RIF_1);
% filtre RII
data_iir = filter(RII_b,RII_a,data);

% réalisation du filtre notch
z1 = exp(1i*0.55*pi);
z2 = conj(z1);
zz = [z1, z2];
rr = 10^((log10(0.001))/1000);
pp = rr.*zz;

[bb,aa] = zp2tf(zz',pp,1);
```

Poles et zeros

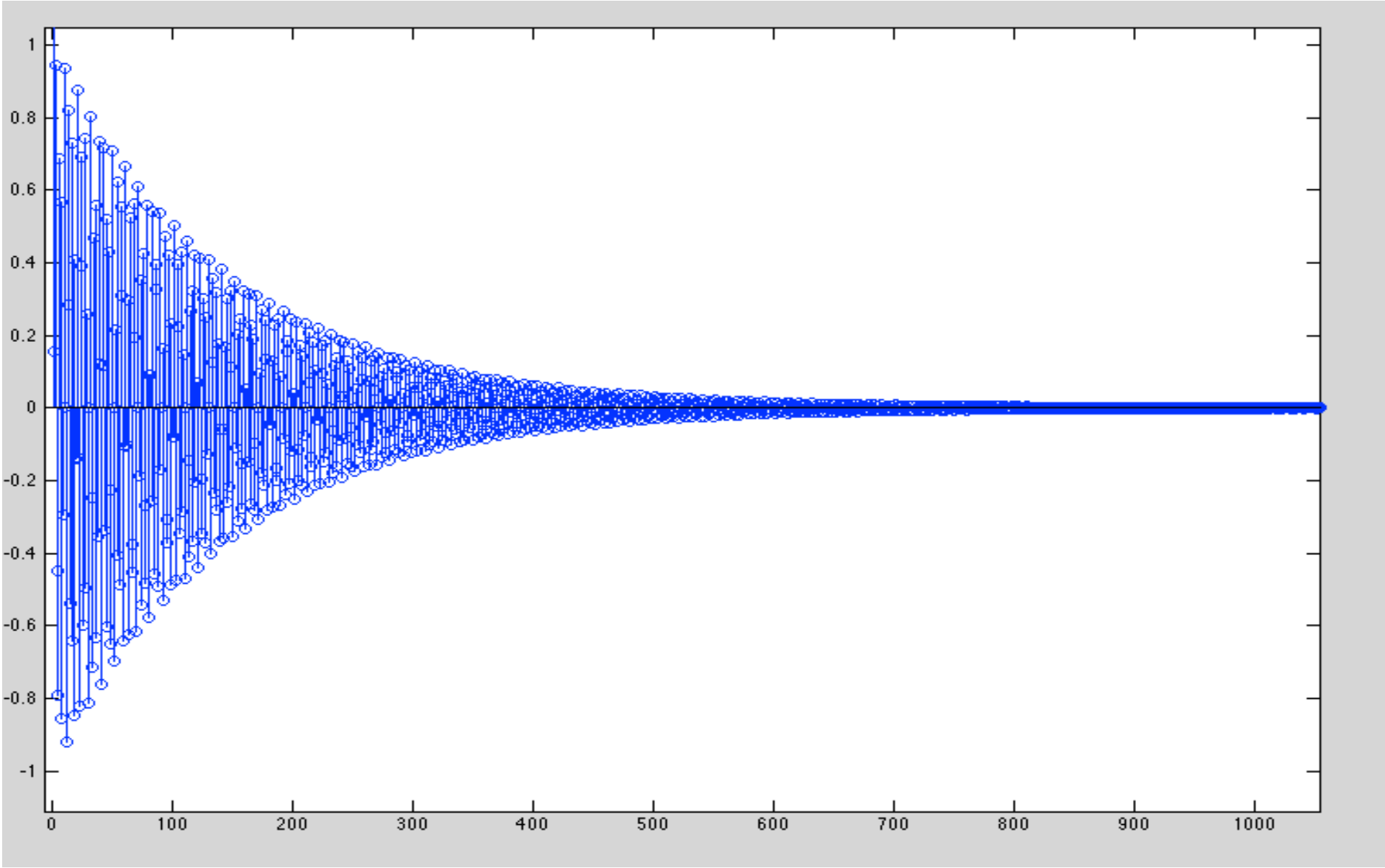


Réponse en fréquence

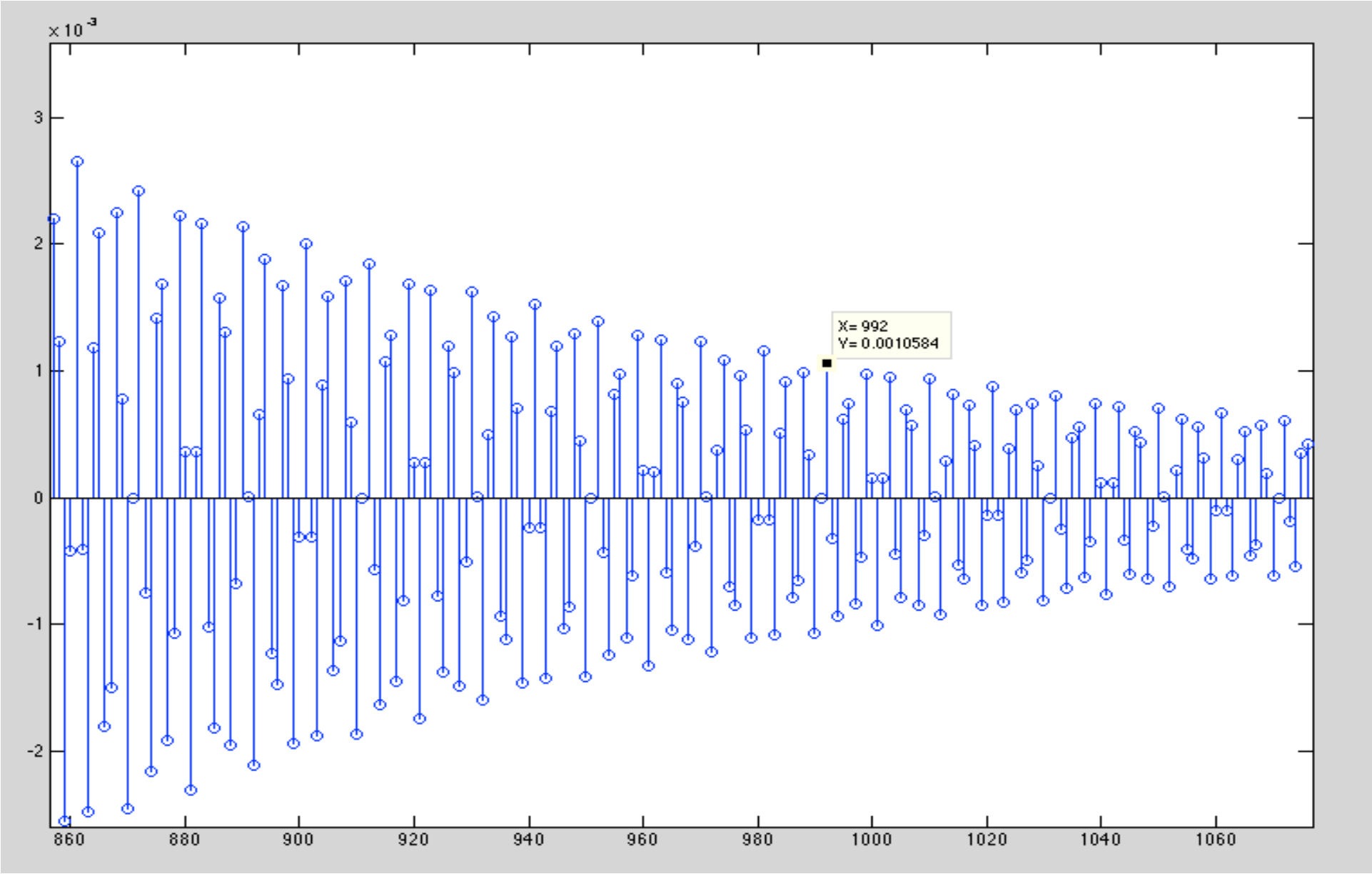




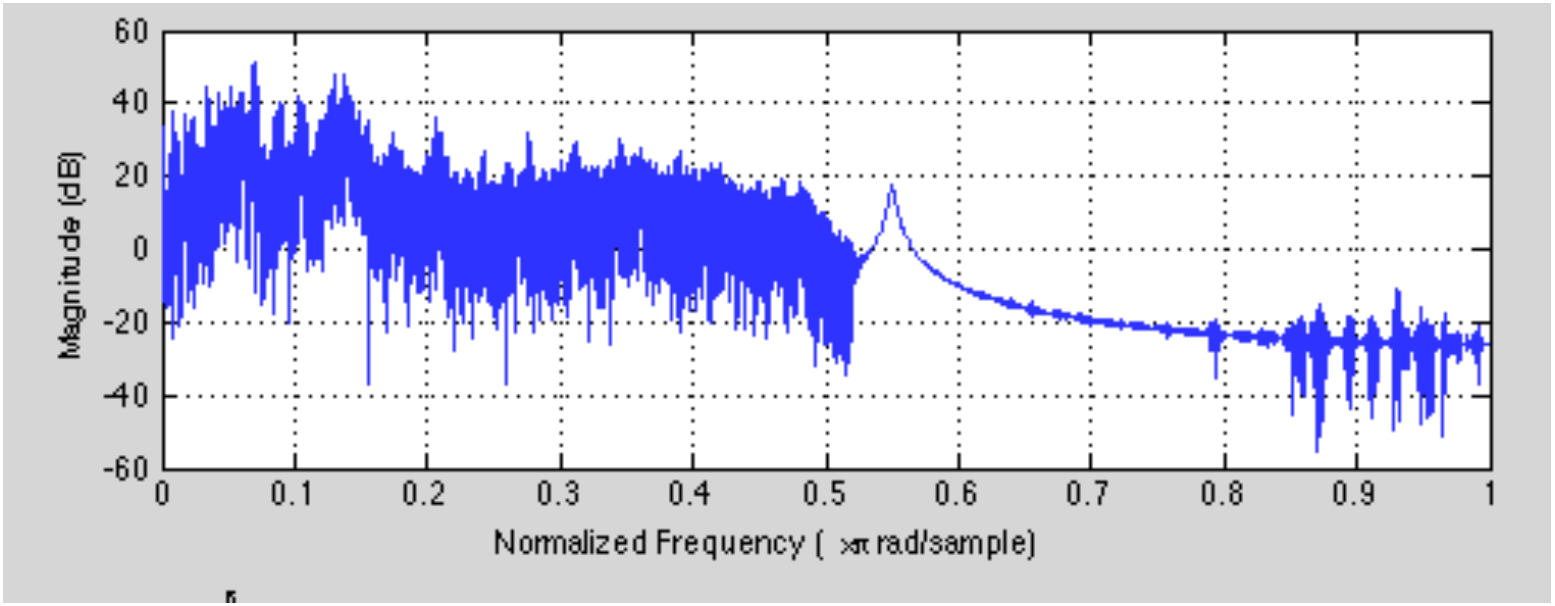
Réponse à l'impulsion du filtre notch (normalisée)

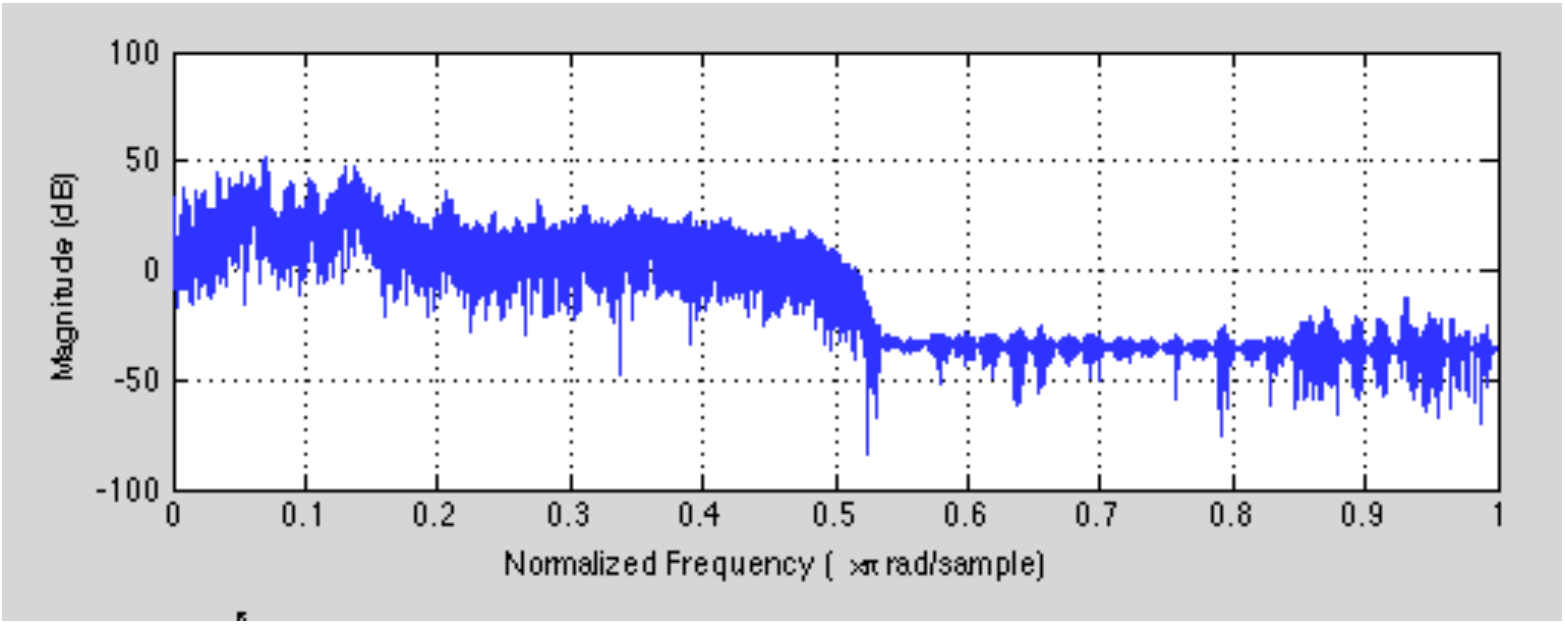


Réponse à l'impulsion du filtre notch (normalisée)

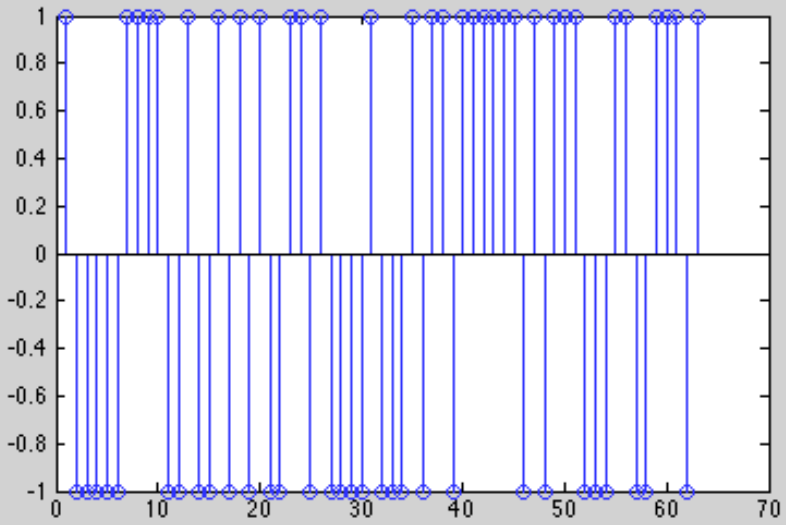


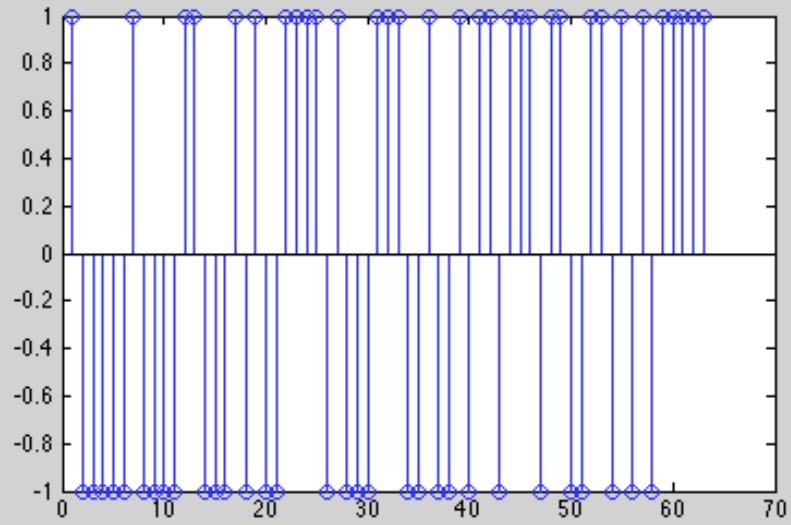
DFT du signal filtré: avec et sans la partie transitoire du signal filtré

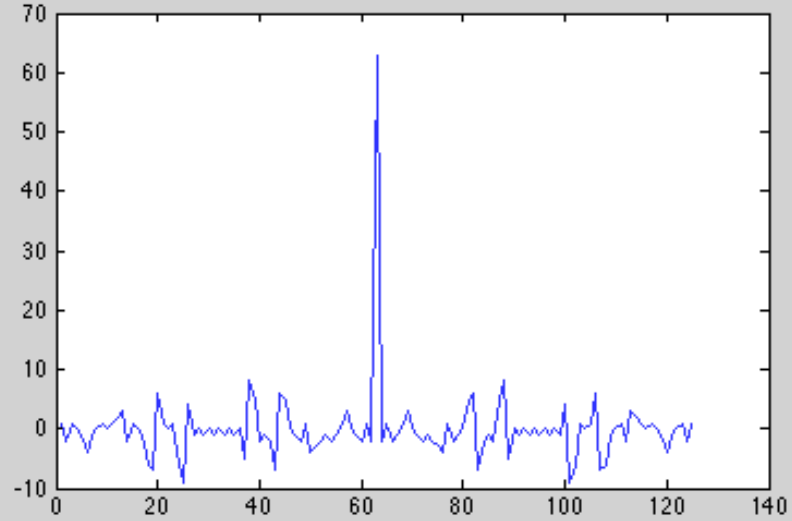


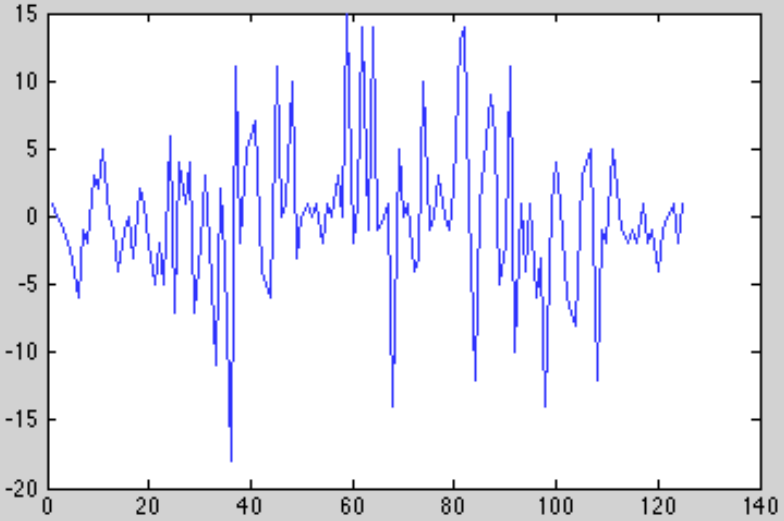


Corrélation de s1 avec s1 et s2



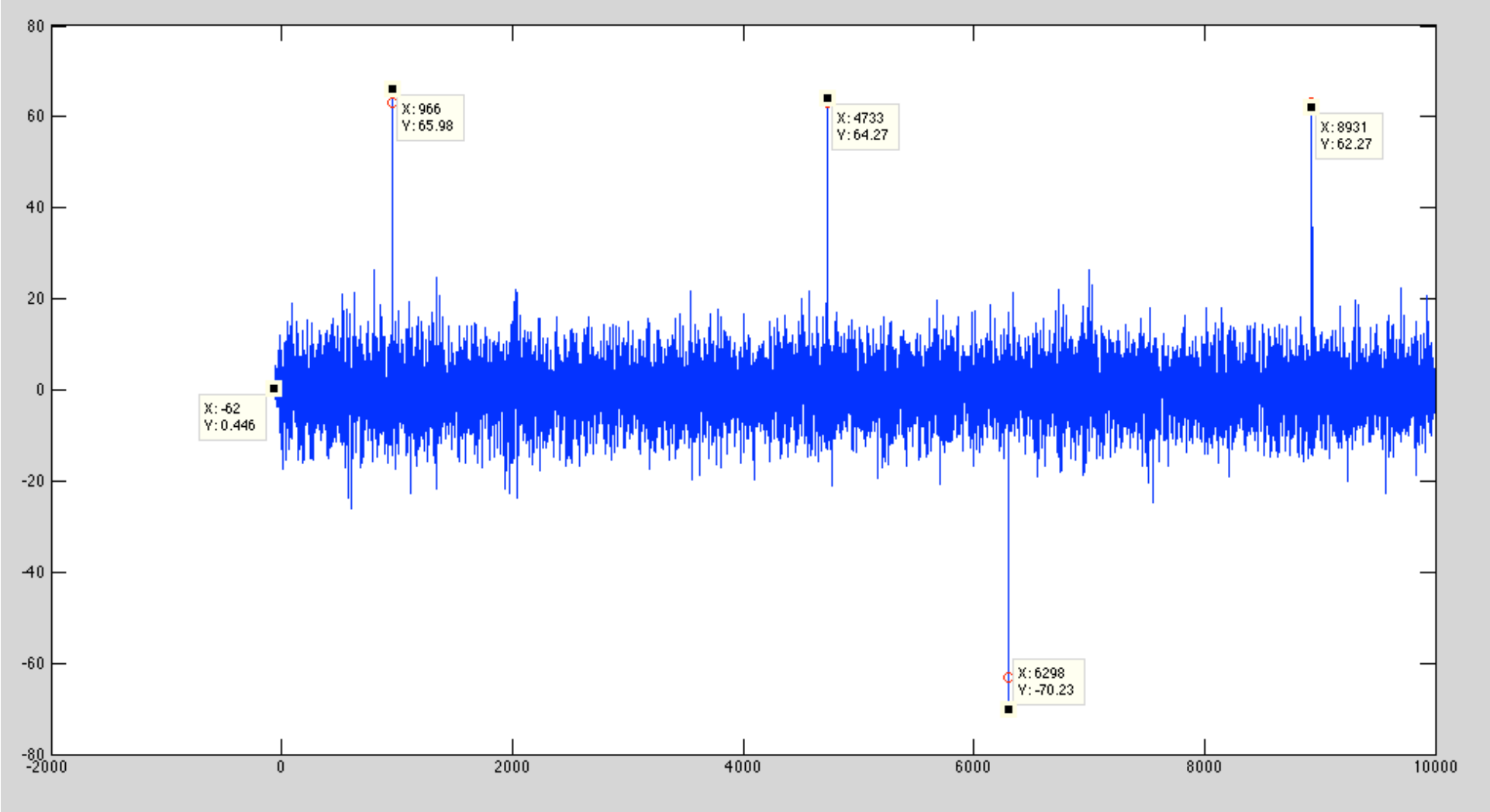






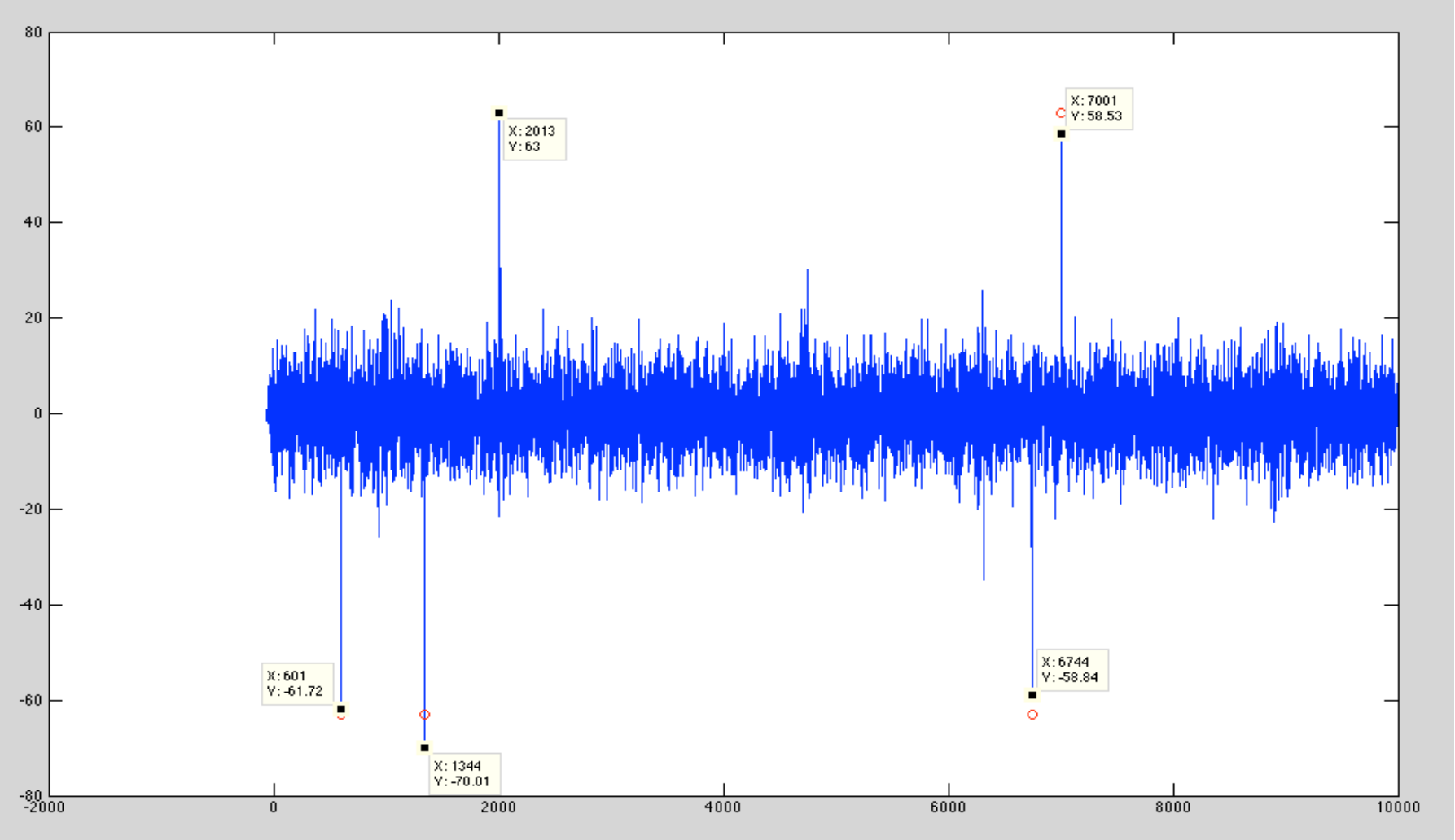
TP #1, Q2

```
out1 = conv(data,fliplr(signal_1));
figure, plot((-62:9999),out1);
```



TP #1, Q2

```
out1 = conv(data,fliplr(signal_2));
figure, plot((-62:9999),out1);
```



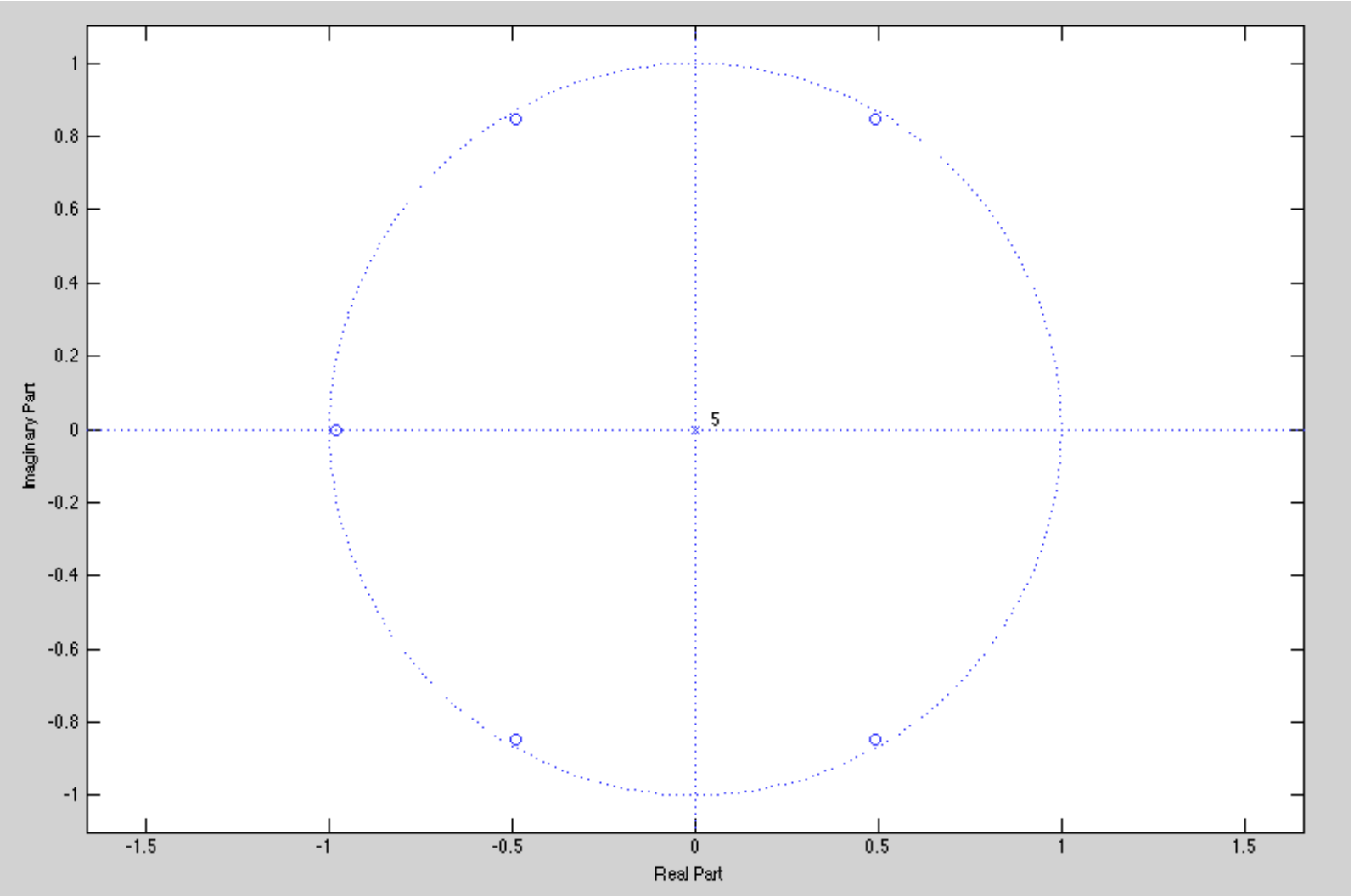
TP #1, Q2 - overlap-add

```
% nb de points du signal de sortie: 1000+63-1
% coordonnées absolues: -62:9999
% coordonnées matlab: 1:10062
out1 = zeros(1,10000+63-1);  %vecteur de 10062 zéros
% signal découpé en segments de 1000
for i=0:9
    data_s = data(i*1000 +1: (i+1)*1000);
    out_s = conv(data_s, fliplr(signal_1));
    % data_s: i*1000 +1: (i+1)*1000;
    % fliplr(signal_1):  -63:-1
    % out_s: +63 - diff en coord. absolues et matlab
    out1(-63+i*1000+1+63:(i+1)*1000-1+63) =
      out1(-63+i*1000+1+63:(i+1)*1000-1+63) + out_s;
end
figure, plot((-62:9999),out1);
```

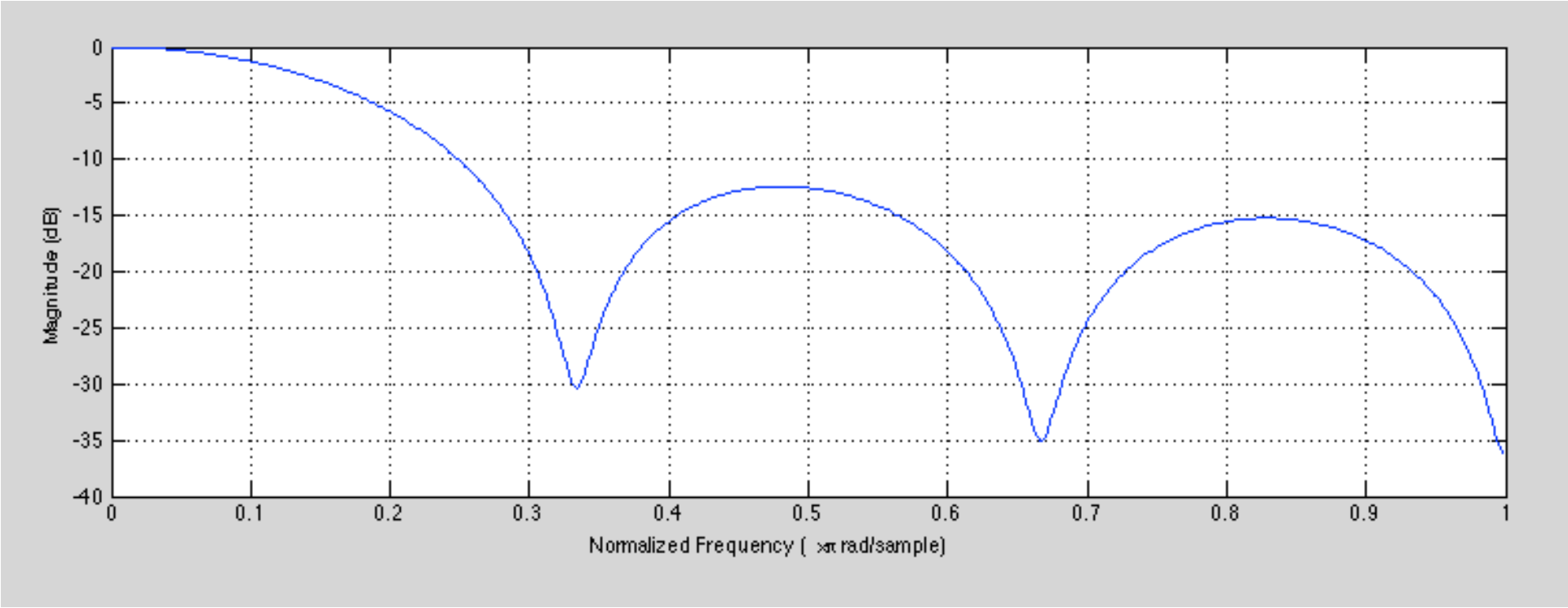
TP #1, Q3

```
% filtre inverse
rii_a = rif1;
rii_b = 1;

% pour obtenir les valeurs des zéros de rif1
abs(roots(rif1)); angle(roots(rif1));

% filtrage d'une colonne de l'image
img_Yr(:,i) = filter(rii_b,rii_a,img1(:,i));
% affichage de l'image
figure, imshow(img_Yr,[0 255])
```

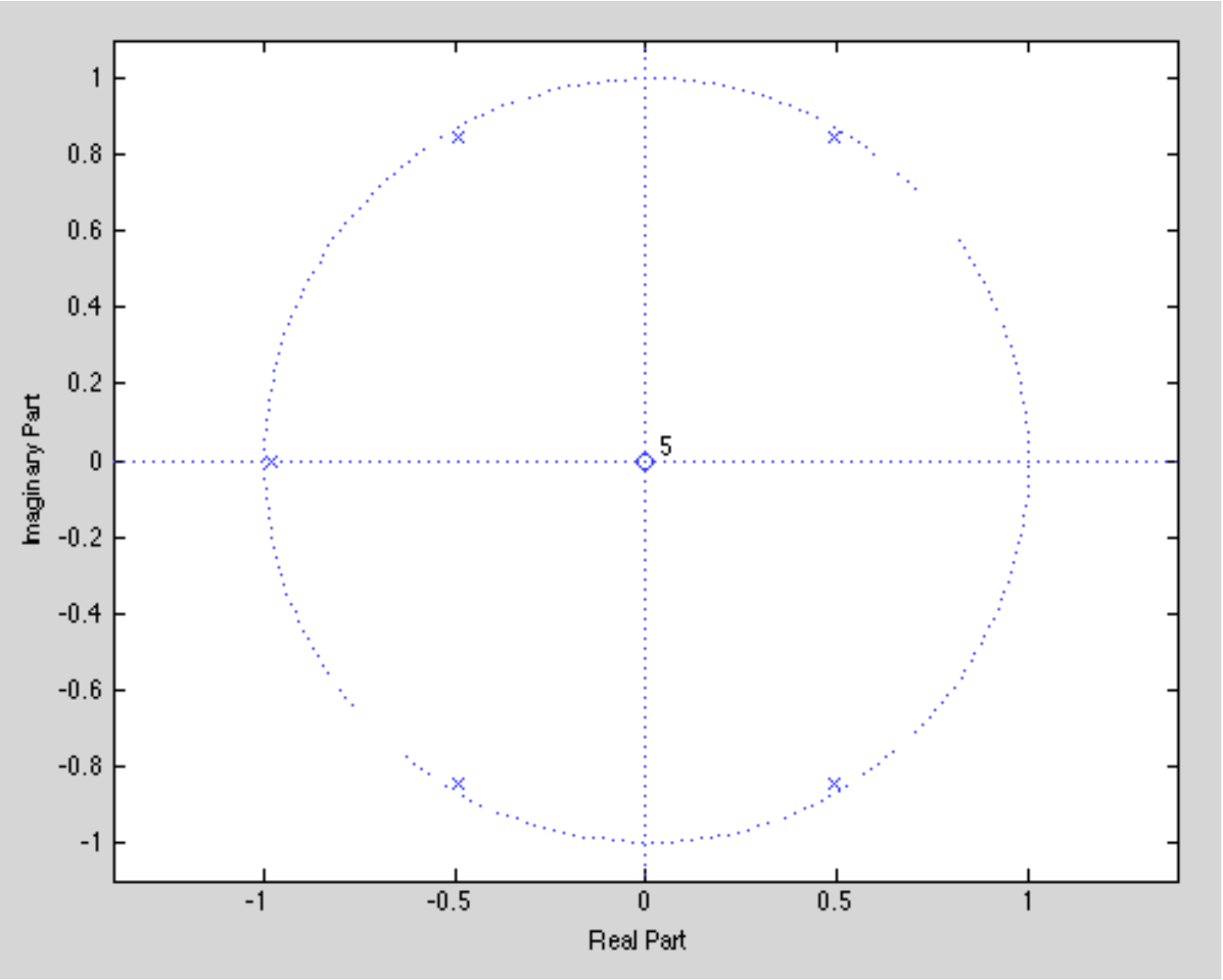
Poles et zéros – filtre RIF1



Réponse en fréquence du filtre RIF1



Poles et zéros – filtre inverse



Réponse en fréquence du filtre inverse

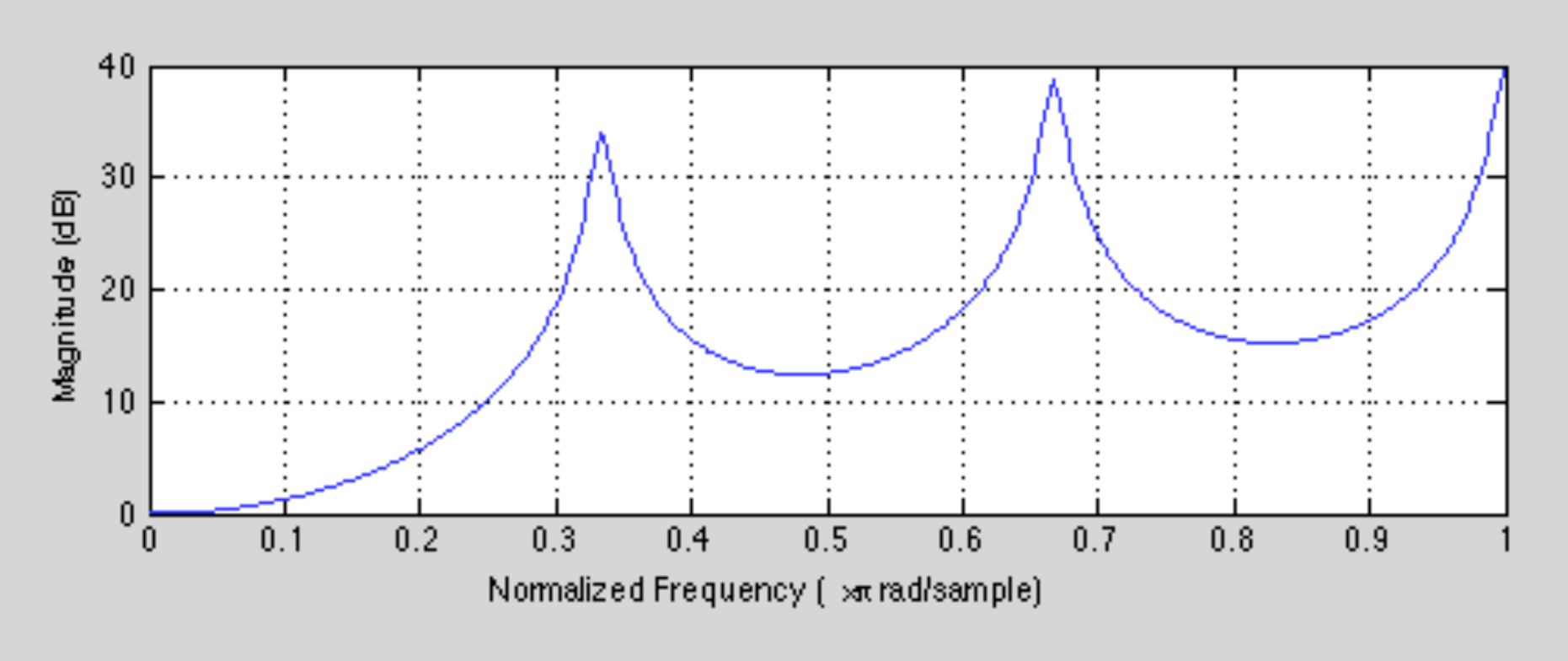
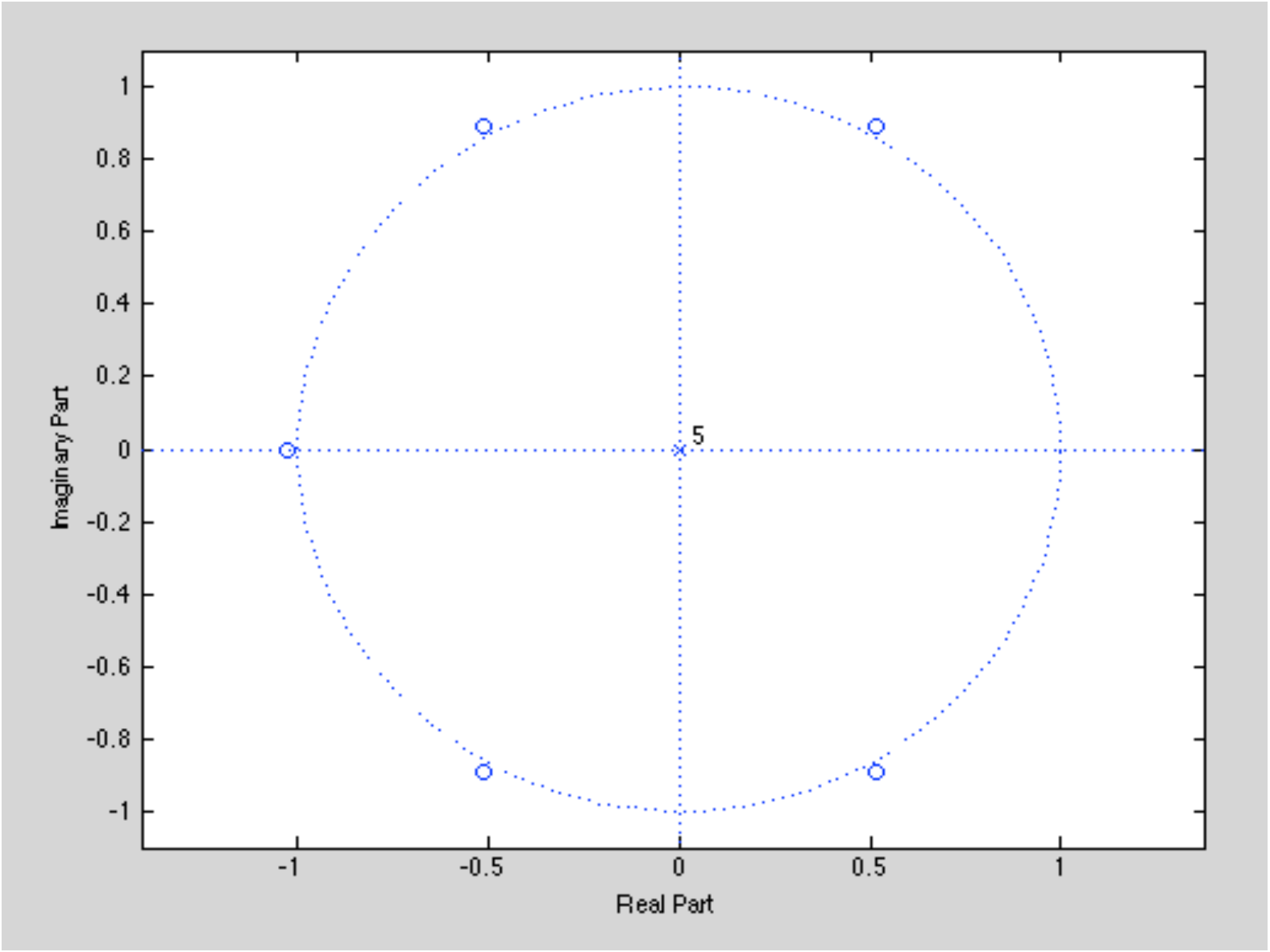
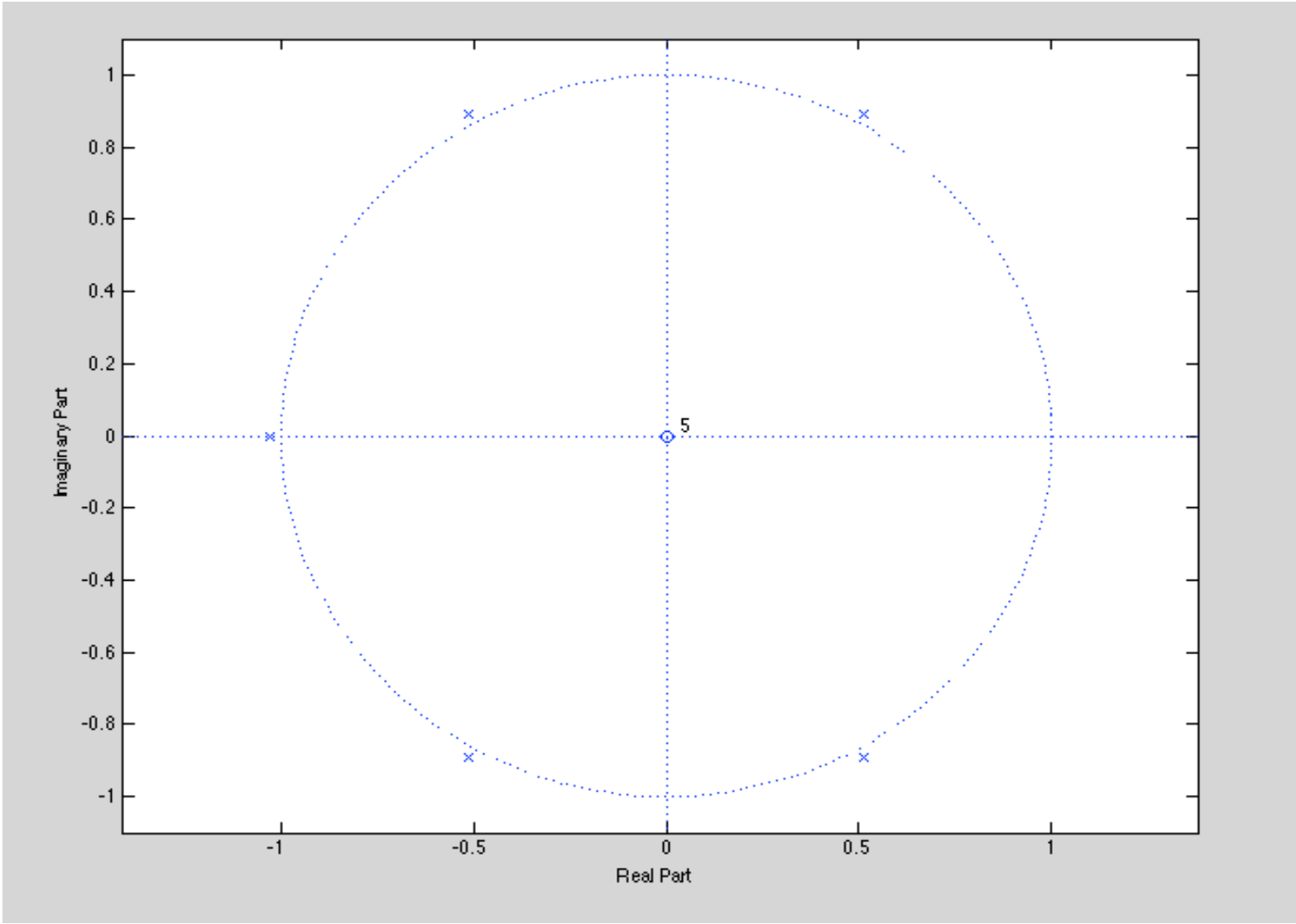


Image sortie

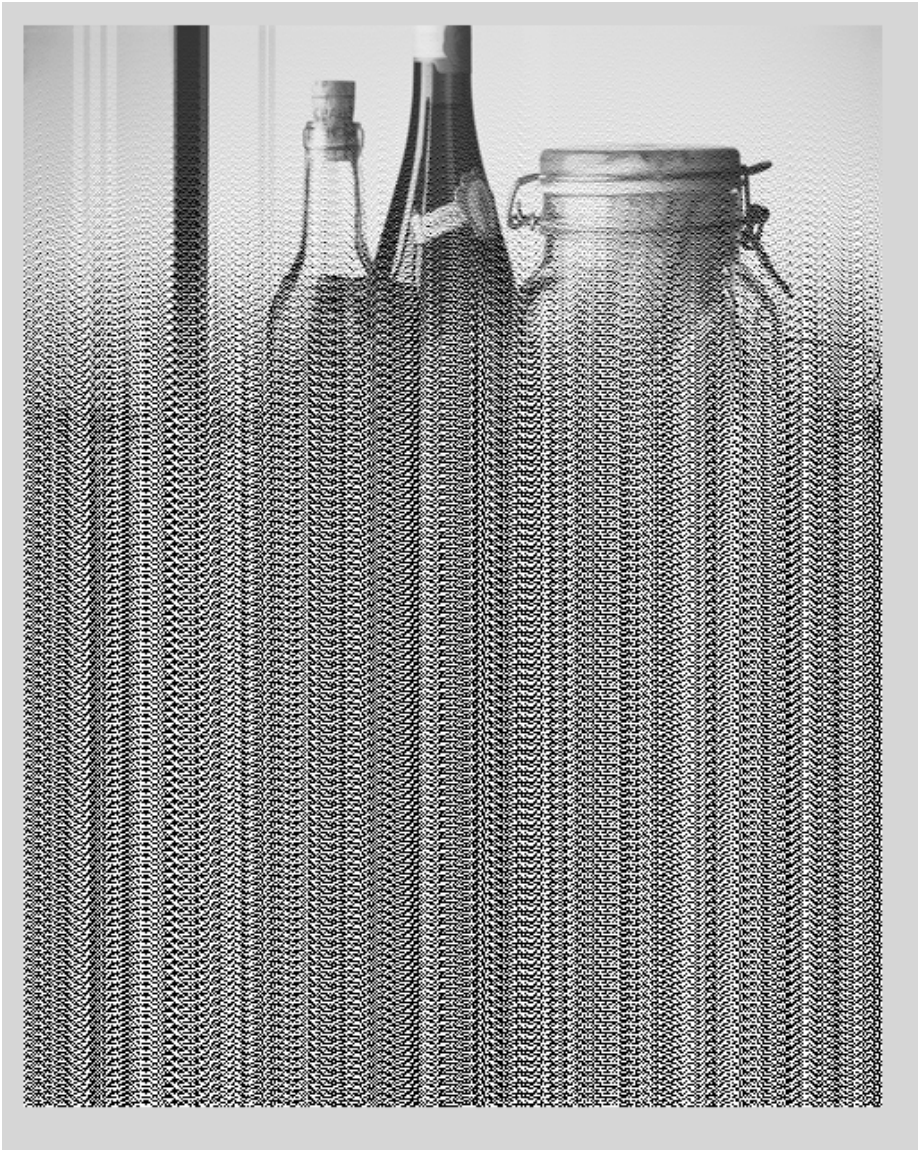
Poles et zéros – filtre RIF2



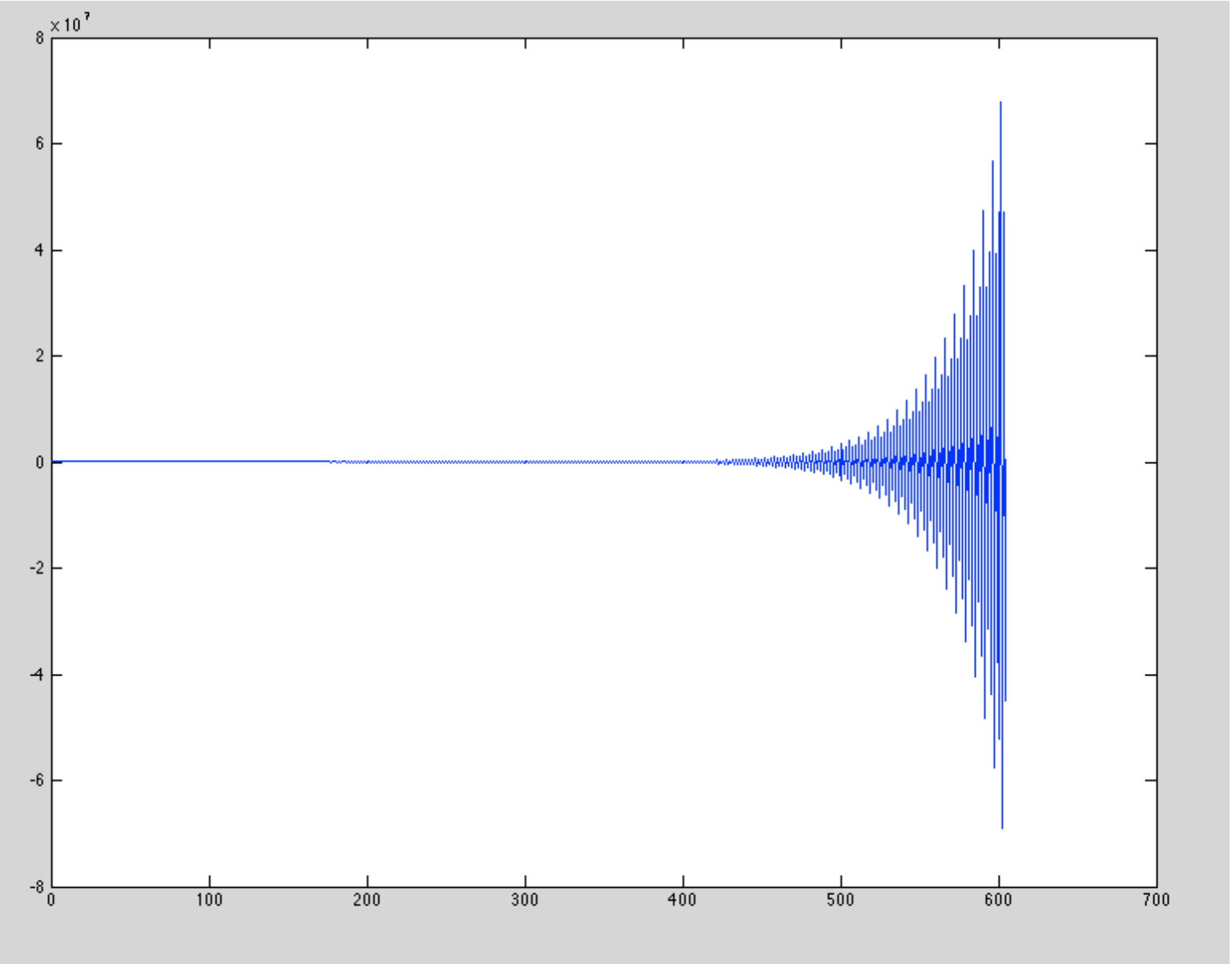
Poles et zéros – filtre inverse



Filtre inverse instable



Une colonne de l'image



TP #1, Q3b)

```
% coefficients "a" du filtre inverse
aa = rif2;


% equation aux diff. à résoudre
% a(0)y(n) + a(1)y(n-1) + a(2)y(n-2) + a(3)y(n-3) +
a(4)y(n-4)+a(5)y(n-5) = delta(n)
% isole y(n-5)
% y(n-5) = -(1/a(5))(a(0)y(n) + a(1)y(n-1) + a(2)y(n-2)
+ a(3)y(n-3) + a(4)y(n-4)) + delta(n)/a(5);


% premier point non nul: y(-5); a(i) -> aa(i+1)
yy = zeros(256,1);  % n=0 -> yy(256); n=-255 -> yy(1)
yy(-5+ 256) = 1/aa(6); % solution pour n=0
```

TP #1, Q3b)

```
% equation aux diff. à résoudre
% y(n-5) = -(1/a(5))(a(0)y(n) + a(1)y(n-1) + a(2)y(n-2)
+ a(3)y(n-3) + a(4)y(n-4)) + delta(n)/a(5);
for n=-1:-1:-249
    yy(n-5 + 256) = -(1/aa(6))*(aa(1)*yy(n+256)+
aa(2)*yy(n-1+256) + aa(3)*yy(n-2+256) + aa(4)*yy(n-3+
256) + aa(5)*yy(n-4+256));
end

figure, stem(yy);

[Li,Co] = size(img2);
for i=1:Co
    img_Yr3(:,i) = conv(img2(:,i),yy);
end
figure, imshow(img_Yr3,[0 255])
```

Réponse à l'impulsion du filtre stable 256 premiers points avec retard de 256

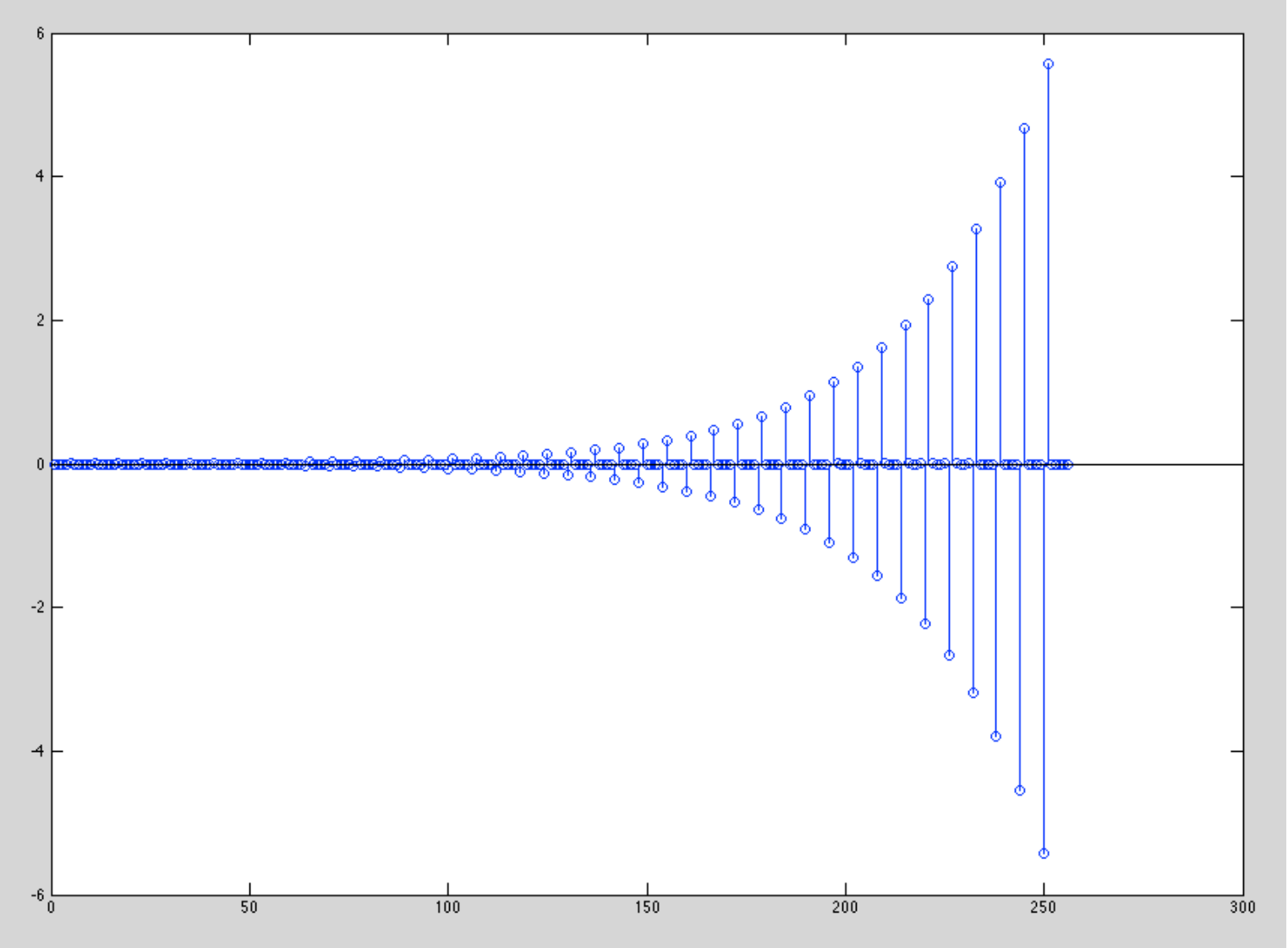
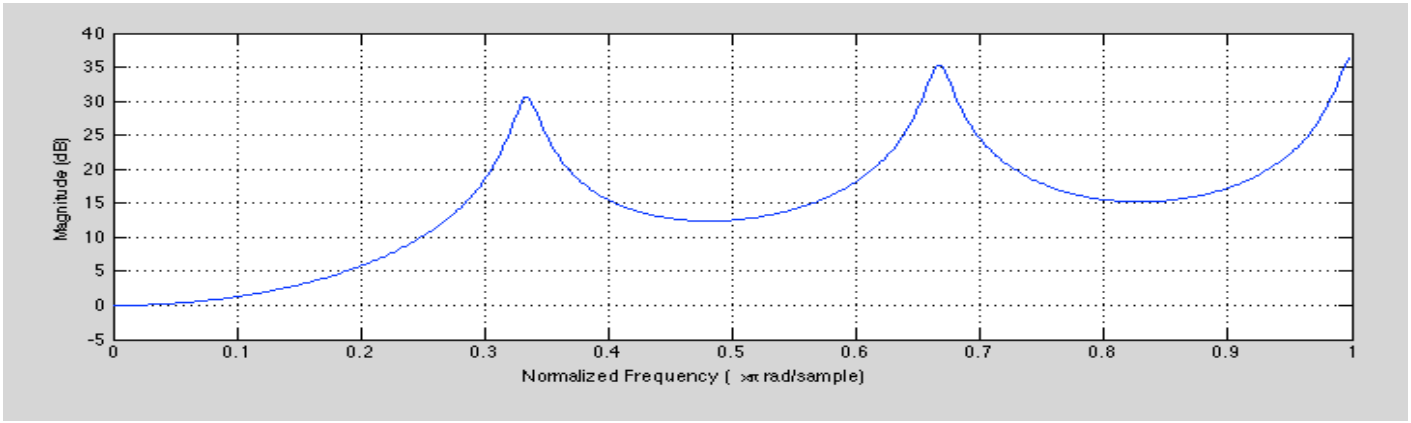
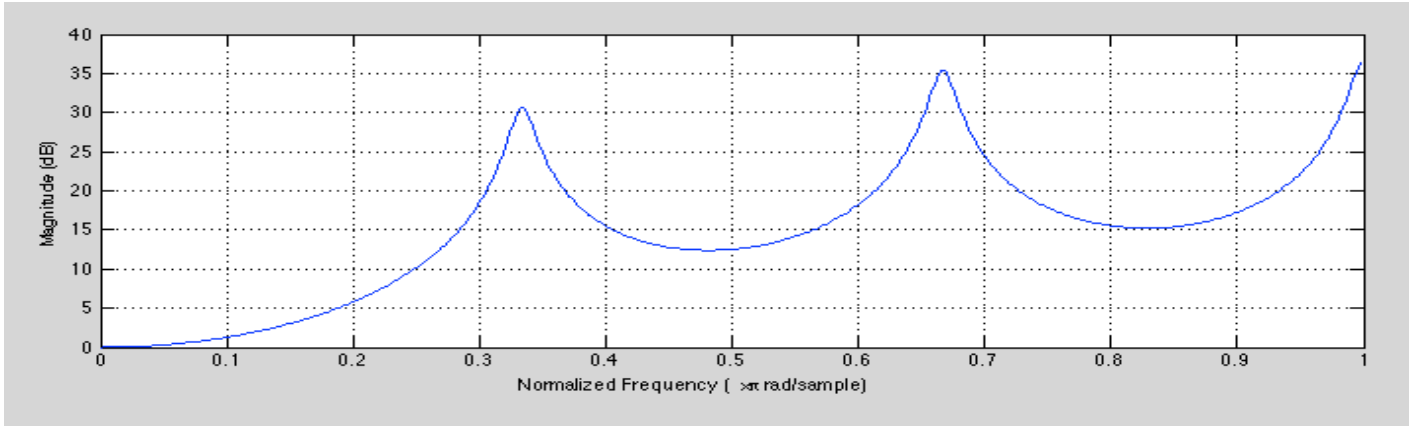


Image filtrée: filtre RIF de 256 pts

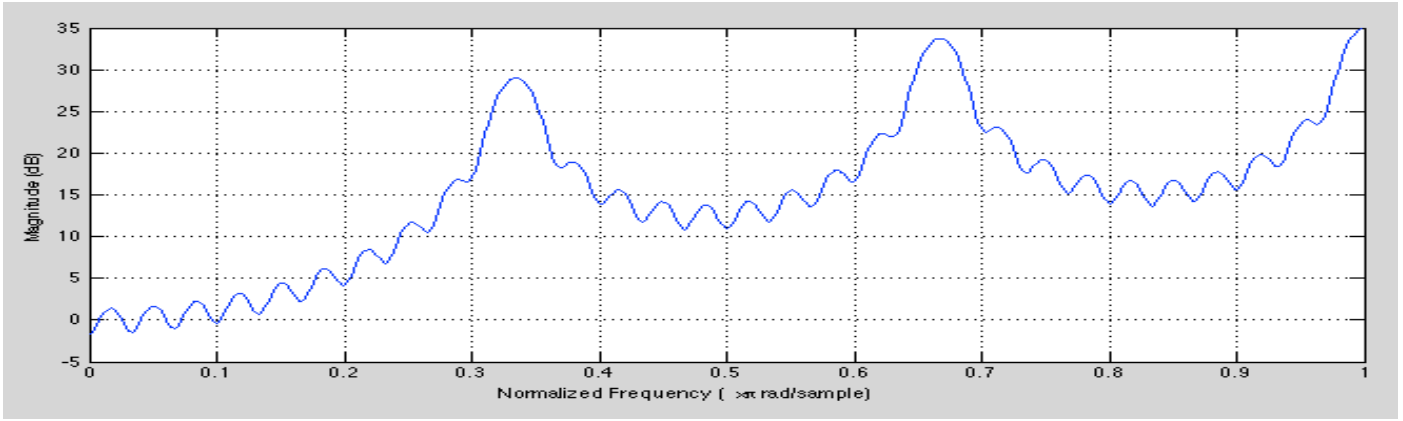
Image filtrée: filtre RIF de 64 pts

Réponses en fréquence des filtres





256 pts



64 pts